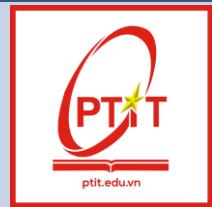


LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

CHƯƠNG 7:

Network Connection

- ❖ Giảng viên: Ths. Nguyễn Trung Hiếu
- ❖ Email:hieunt.tg@ptithcm.edu.vn
- ❖ Mobie: 0983051825



Phần I: JSON

-  Giới thiệu về JSON
-  JSON Object và JSON Array

Phần II: Volley Android

-  Giới thiệu về Volley
-  Tạo Database và Table

1

JSON

- JSON là chữ viết tắt của Javascript Object Notation, đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được, bạn có thể sử dụng lưu nó vào một file, một record trong CSDL rất dễ dàng. JSON có định dạng đơn giản, dễ dàng sử dụng và truy vấn hơn XML rất nhiều nên tính ứng dụng của nó hiện nay rất là phổ biến.
 - JSON là viết tắt của JavaScript Object Notation
 - JSON là định dạng trao đổi dữ liệu văn bản dung lượng nhẹ
 - JSON là ngôn ngữ độc lập
 - JSON được "tự mô tả" và dễ hiểu

Download Android Studio X New Tab X PMA localhost:8080 / 127.0.0.1 X localhost:8080/androidse X localhost:8080/androidse X JF Best JSON Formatter and X V Tim hiểu về chuỗi dữ liệu X + - ☰

https://jsonformatter.org

{JSON formatter}

JSON BEAUTIFIER JSON PARSER XML FORMATTER JSBEAUTIFIER SAVE RECENT LINKS LOGIN

Sample Upload Data

Validate

2 Tab Space

Format / Beautify

SQUARESPACE
EMAIL CAMPAIGNS LET YOU BUILD A COMMUNITY START YOUR FREE TRIAL

Minify / Compact

Convert JSON to-

Download

Code

1 ["HoTen": "Ph\u01ea1m Y\u01ebfn Th\u00e0nh\r\n", "MaSV": "1911060259", "Email": "pyt312001@gmail.com \r\n"], { "HoTen": "V\u00e0\u00f5f5 C\u00e1\u00f4ng H\u01ed3\r\n", "MaSV": "1911062012", "Email": "adadad077@gmail.com\r\n"}, { "HoTen": "Tr\u01ea7n Hu\u01ef3nh Ng\u01ecdc H\u00e0\u00e2n\r\n", "MaSV": "1911065207", "Email": "Tranduydang193@gmail.com\r\n"}, { "HoTen": "Nguy\u01ec5n C\u00e1\u00f4ng Danh\r\n", "MaSV": "1911060360", "Email": "congdanh785@gmail.com\r\n"}]

1 {
2 "HoTen": "Phạm Yên Thành\r\n",
3 "MaSV": "1911060259",
4 "Email": "pyt312001@gmail.com \r\n"
5 },
6 {
7 "HoTen": "Võ Công Hồ\r\n",
8 "MaSV": "1911062012",
9 "Email": "adadad077@gmail.com\r\n"
10 },
11 {
12 "HoTen": "Trần Huỳnh Ngọc Hân\r\n",
13 "MaSV": "1911065207",
14 "Email": "Tranduydang193@gmail.com\r\n"
15 },
16 {
17 "HoTen": "Nguyễn Công Danh\r\n",
18 "MaSV": "1911060360",
19 "Email": "congdanh785@gmail.com\r\n"
20 }
21]

- Chuỗi JSON được bao lại bởi dấu ngoặc nhọn {}
- Các key, value của JSON bắt buộc phải đặt trong dấu nháy kép "{}".
- Nếu có nhiều dữ liệu (nhiều cặp key => value) thì ta dùng dấu phẩy (,) để ngăn cách

JSON OBJECT

```
1 {  
2   "HoTen": "Phạm Yến Thành\r\n",  
3   "MaSV": "1911060259",  
4   "Email": "pyt312001@gmail.com \r\n"  
5 },  
6 {  
7   "HoTen": "Võ Công Hồ\r\n",  
8   "MaSV": "1911062012",  
9   "Email": "adadad077@gmail.com\r\n"  
10 },  
11 {  
12   "HoTen": "Trần Huỳnh Ngọc Hân\r\n",  
13   "MaSV": "1911065207",  
14   "Email": "Tranduydang193@gmail.com\r\n"  
15 },  
16 {  
17   "HoTen": "Nguyễn Công Danh\r\n",  
18   "MaSV": "1911060360",  
19   "Email": "congdanh785@gmail.com\r\n"  
20 }  
21 }  
22 }
```

Object trong Json được thể hiện bằng dấu ngoặc nhọn {}. Khái niệm Object trong Json cũng khá tương đồng với Object trong Javascript. Tuy nhiên, Object trong Json vẫn có những giới hạn như:

- **Key:** phải luôn nằm trong dấu ngoặc kép, không được phép là biến số.
- **Value:** Chỉ cho phép các kiểu dữ liệu cơ bản: numbers, String, Booleans, arrays, objects, null. Không cho phép function, date, undefined.
- Không cho phép dấu phẩy cuối cùng như Object trong Javascript.

JSON ARRAY

```
1 [ ]  
2 {  
3     "HoTen": "Phạm Yến Thành\r\n",  
4     "MaSV": "1911060259",  
5     "Email": "pyt312001@gmail.com \r\n"  
6 },  
7 {  
8     "HoTen": "Võ Công Hồ\r\n",  
9     "MaSV": "1911062012",  
10    "Email": "adadad077@gmail.com\r\n"  
11 },  
12 {  
13     "HoTen": "Trần Huỳnh Ngọc Hân\r\n",  
14     "MaSV": "1911065207",  
15     "Email": "Tranduydang193@gmail.com\r\n"  
16 },  
17 {  
18     "HoTen": "Nguyễn Công Danh\r\n",  
19     "MaSV": "1911060360",  
20     "Email": "congdanh785@gmail.com\r\n"  
21 }  
22 [ ]
```

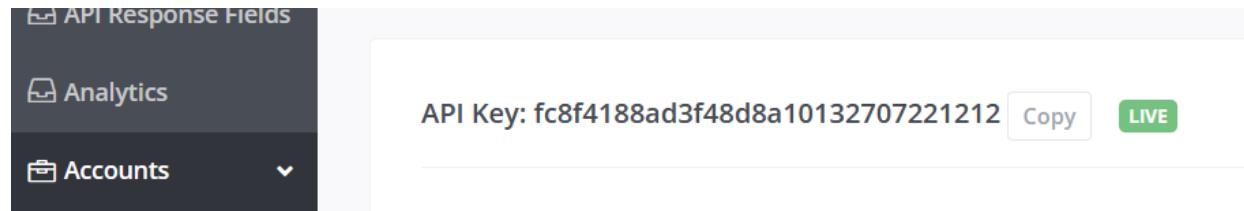
JSON Array là danh sách thứ tự các giá trị. Mảng trong JSON có thể lưu trữ nhiều giá trị. Nó có thể lưu trữ chuỗi, số, boolean hoặc đối tượng trong mảng JSON. Trong mảng JSON, các giá trị phải được phân tách bằng dấu phẩy.

[] (Dấu ngoặc vuông) đại diện cho mảng JSON.

Vào trang :

http://weatherapi.com

Đăng ký 1 tài khoản – Login



Call

```
https://api.weatherapi.com/v1/forecast.json?key=fc8f4188ad3f48d8a10132707221212&q=London&days=1&aqi=no&alerts=no
```

2

Volley Android

- Volley library Android là thư viện networking cho các dự án Android. Volley được dùng để quản lý các network request, giúp cho developer đơn giản hóa việc thực hiện kết nối và xử lý kết quả trả về từ server.
- Volley hỗ trợ đầy đủ các HTTP request như GET, POST, PUT, DELETE. Ngoài ra, các thư viện network kiểu như Volley có tính năng network cache rất hữu ích.
- Tất cả request được cache nhằm đảm bảo khi reload, sẽ có kết quả nhanh hơn. Ngoài ra, response được lưu trong memory nên với các response có dữ liệu như JSON, Image, String... thì rất hiệu quả.

Một số tính năng nổi bật của Volley:

- Hỗ trợ lên schedule để tạo request
- Cho phép cùng lúc thực hiện nhiều request trên các thread khác nhau theo độ ưu tiên.
- Cache trên Disk hay RAM
- Cho phép hủy một request.
- Dễ dàng tùy chỉnh cho phù hợp với yêu cầu của ứng dụng. Như thiết lập retry, back off.
- Dễ debug.

- ✓ 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 **Sync Now** để Android Studio download và nạp thư viện vào project.

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

Sử dụng thư viện Volley library Android

1. Thêm dependencies vào build.gradle

```
dependencies {  
    ...  
    compile 'com.android.volley:volley:1.1.0'  
}
```

2. Thêm permission INTERNET vào manifest nhé (permission này không cần phải tạo request run time tới người dùng)

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

- ✓ 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.

Sử dụng Send a simple request

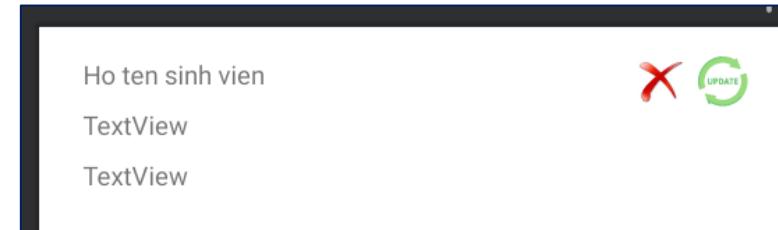
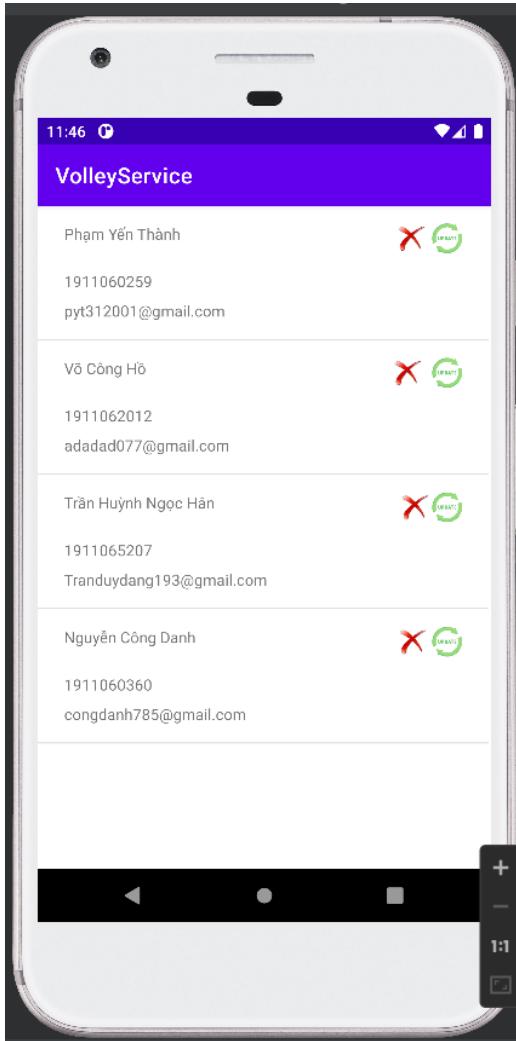
```
final TextView textView = (TextView) findViewById(R.id.text);
// ...

// Instantiate the RequestQueue.
RequestQueue queue = Volley.newRequestQueue(this);
String url = "https://www.google.com";

// Request a string response from the provided URL.
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        // Display the first 500 characters of the response string.
        textView.setText("Response is: " + response.substring(0,500));
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        textView.setText("That didn't work!");
    }
});

// Add the request to the RequestQueue.
queue.add(stringRequest);
```

VOLLEY ANDROID



```
public class InforSv {  
    private int ID;  
    private String hoTen;  
    private String maSV;  
    private String email;  
  
    public InforSv( String hoTen, String maSV, String email) {  
        this.ID = ID;  
        this.hoTen = hoTen;  
        this.maSV = maSV;  
        this.email = email;  
    }  
    getter/setter  
}
```

VOLLEY ANDROID

```
public class inforSVAdapter extends BaseAdapter {  
    private Context context;  
    private int layout;  
    private List<InforSv> inforSVList ;  
  
    public inforSVAdapter(Context context, int layout, List<InforSv> inforSVList) {  
        this.context = context;  
        this.layout = layout;  
        this.inforSVList = inforSVList;  
    }  
    ...  
    private class ViewHolder{  
  
        TextView txtHoten, txtMaSV, txtEmail;  
        ImageView imgDelete, imgEdit;  
    }  
    @Override  
    public View getView(int i, View view, ViewGroup viewGroup) {  
        ViewHolder holder;  
        if (view == null)  
        {  
            holder = new ViewHolder();  
            LayoutInflater inflater = (LayoutInflater)  
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
            view = inflater.inflate(layout,null);  
            holder.txtHoten = view.findViewById(R.id.txtHoten);  
            holder.txtMaSV = view.findViewById(R.id.txtMasv);  
            holder.txtEmail = view.findViewById(R.id.txtemail);  
            holder.imgDelete = view.findViewById(R.id.imgDelete);  
            holder.imgEdit = view.findViewById(R.id.imgEdit);  
            view.setTag(holder);  
        }  
        else  
        {  
            holder = (ViewHolder) view.getTag();  
        }  
        InforSv inforSv = inforSVList.get(i);  
        holder.txtHoten.setText(inforSv.getHoTen());  
        holder.txtMaSV.setText(inforSv.getMasv());  
        holder.txtEmail.setText(inforSv.getEmail());  
  
        holder.imgDelete.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast.makeText(view.getContext(),inforSv.getHoTen(),Toast.LENGTH_SHORT).show();  
            }  
        });  
  
        return view;  
    }  
}
```

```
public class MainActivity extends AppCompatActivity {
```

```
    String link="http://192.168.1.3:8080/androidservice/getdata.php";
    ListView lvInforSV;
    ArrayList<InforSv> inforSvArrayList;
    inforSvAdapter SVAdapter ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvInforSV = findViewById(R.id.lvInforSV);
        inforSvArrayList = new ArrayList<>();
        SVAdapter = new
        inforSvAdapter(this,R.layout.dong_thong_tin_sv,inforSvArrayList);
        lvInforSV.setAdapter(SVAdapter);
        GetData(link);
    }
```

VOLLEY ANDROID

```
private void GetData(String url) {  
    RequestQueue requestQueue = Volley.newRequestQueue(this);  
    JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(Request.Method.GET, url, null,  
        new Response.Listener<JSONArray>() {  
            @Override  
            public void onResponse(JSONArray response) {  
                Log.d("AAA",response.toString());  
                for (int i=0;i<response.length();i++)  
                {  
                    try {  
                        JSONObject object = response.getJSONObject(i);  
                        inforSvArrayList.add(new InforSv(object.getString("HoTen"),  
                            object.getString("MaSV"),  
                            object.getString("Email")));  
                    } catch ( JSONException e ) {  
                        e.printStackTrace();  
                    }  
                }  
                SVAdapter.notifyDataSetChanged();  
            }  
, new Response.ErrorListener() {  
            @Override  
            public void onErrorResponse(VolleyError error) {  
                Toast.makeText(MainActivity.this,"Loi",Toast.LENGTH_LONG).show();  
                Log.d("AAA",error.toString());  
            }  
});  
requestQueue.add(jsonArrayRequest);  
}
```

```
public class MainActivity extends AppCompatActivity {
```

