Volley

Volley là 1 thư viện network được phát triển bởi google. Giới thiệu ở Google I/O 2013.

Tại sao dùng volley. HttpUrlConnection và HttpClient đã không còn hoàn hảo, nó phát sinh một số lỗi vẫn chưa được fix. HttpClient cũng đã bị dự định thay đổi trong API 22.

AsynTask cũng rất hay nhưng nó luôn luôn là FIFO. Asyntask cũng phát sinh vấn đề khi danh sách load từ internet, khi ta kéo xuống, ta cũng không nói với Asyntask nạp trang tiếp theo được. Vd giống facebook.

Volley giải quyết nhiều vấn đề với API giản ước. Ta không cần kiểm tra onPostExecute khi activity bị huỷ, nó giúp cho ta tránh lỗi NullPointException.

Nó nhanh hơn, tự động cache mọi thứ

Nó làm việc trên 3 level, mỗi level thao tác trên thread riêng của nó.



Trong main thread là các thứ ta làm với Asyntask , không hơn không kém. Với nó ta sẽ không cần doInBackground, nó tự động quản lý các thao tác HTTP và bắt các lỗi về network.

Khi ta thêm một request vào queue, nó sẽ kiểm tra request có thể được cached phục vụ hay không từ đó có sự chọn lựa để phục vụ cho tốt hơn.

Trong network thread một loạt thao tác được thực hiện như tạo các HTTP request, xử lý reponse, viết vào cache...

# Cài đặt Volley

1. Tạo project.
2. Nhúng

compile **'com.mcxiaoke.volley:library:1.0.19'**

hoặc xem bảng mới nhất tại: <https://github.com/mcxiaoke/android-volley>

# Tạo webserver

1. Tạo trang php ví dụ tên untitled.php có mã như sau:

<?php

echo "hello teo";

?>

# Sử dụng Volley

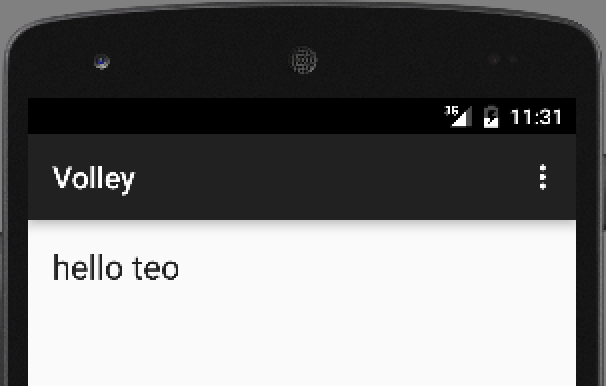
1. Quay về android. Vào androidmanifrest.xml cấp quyền internet.

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

1. Trên giao diện kéo vào một TextView, trong onCreate ánh xạ textview và viết như sau (chú ý trong android dùng 10.0.2.2 để truy xuất đến localhost) :

String url=**"http://10.0.2.2/testios/untitled.php"**;  
StringRequest stringrequest=**new** StringRequest(  
 Request.Method.***GET***, url, **new** Response.Listener<String>() {  
 @Override  
 **public void** onResponse(String response) {  
 **tv**.setText(response);  
 }  
}, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError error) {  
  
 }  
});  
Volley.*newRequestQueue*(**this**).add(stringrequest);

1. Chạy ct xem kết quả hiện lên textview.



Công việc HTTP và asyntask được giải quyết bằng 1 đoạn mã rất gọn. Oh my ăn roi

Volley hầu như chỉ làm việc trên 2 class RequestQueue và Request.

Ta cần tạo ra RequestQueue nó quản lý các thread và đưa dữ liệu ngược về cho main thread. Sau đó ta có thể tạo ra một hay nhiều Request và đưa vào RequestQueue.

Hàm tạo của Request nhận vào các đối số : phương thức Get hay Post, địa chỉ, các event listener.

Hàm onResponse giống như onPostExecute dùng cập nhật dữ liệu lên giao diện.

# POST & GET

Get thì đơn giản là ghép vào chuỗi URL còn POST thì phải override lên hàm getParam

1. Sửa trang web untitled.php lại như sau:

<?php

$ten=$\_POST["ten"];

$tuoi=$\_POST["tuoi"];

echo "hello ". $ten . " tuoi " .$tuoi;

?>

1. Trong onCreate sửa phương thức lại thành Request.Method.POST. Sau đó override lên hàm getParams

String url = "http://10.0.2.2/testios/untitled.php";  
StringRequest stringrequest = new StringRequest(  
 Request.Method.*POST*, url, new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
 tv.setText(response);  
 }  
}, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
  
 }  
}  
) {  
  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String,String> param=new HashMap<String,String>();  
 param.put("ten","nguyen van teo");  
 param.put("tuoi","5");  
 return param;*//super.getParams();* }  
};  
Volley.*newRequestQueue*(this).add(stringrequest);

Chú ý cấp của hàm getParam. Nhìn hình để thấy rõ hơn



1. Chạy ct, xem kết quả của POST.

# Loại Request

Volley thường dùng 3 loại request chính là

* StringRequest: để lấy về chuỗi
* ImageRequest: để lấy về hình
* JsonRequest, JsonArrayRequest: để lấy về đối tượng json

# JsonRequest, JsonArrayRequest.

ở trên ta đã làm StringRequest rồi, giờ ta làm đến jsonrequest

1. Trên webserver tạo một trang mới vd tên simplejsonarray.php và có nội dung như sau:

<?php

echo'{

"contacts": [

{

"id": "1",

"name": "teo"

},

{

"id": "2",

"name": "ti"

},

{

"id": "3",

"name": "tung"

}

]

}'

?>

(mã json này tuy là array nhưng đối tượng đầu tiên là contacts là object, phải lấy object đến nó rồi mới lấy jsonarray con được).

1. Trong onCreate sửa lại như sau:

String url=**"http://10.0.2.2/testios/simplejsonarray.php"**;  
JsonObjectRequest request=**new** JsonObjectRequest(url, **new** Response.Listener<JSONObject>() {  
 @Override  
 **public void** onResponse(JSONObject response) {  
 **tv**.setText(**"vao roi"**);  
 **try** {  
 JSONArray contacts=(JSONArray)response.getJSONArray(**"contacts"**);  
 *//lay ra thang dau tien xem choi thoi* JSONObject mot\_contact=(JSONObject)contacts.get(0);  
 String id=(String)mot\_contact.get(**"id"**);  
 String name=(String)mot\_contact.get(**"name"**);  
 **tv**.setText(**"id:"**+id + **" name:"**+name);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
}, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError error) {  
  
 }  
});  
Volley.*newRequestQueue*(**this**).add(request);

1. Chạy ct và xem kết quả lấy ra được thằng đầu tiên

