网络

两种方式进行Htttp通信：

1，HttpURLConnection

2，HttpClient

Volley：

适用范围：通信较为频繁，数据量不大。

配置Volley：

1，编译jar包，导入工程使用。

2，compile 'com.mcxiaoke.volley:library:1.0.19'导入使用。

使用：

1，创建一个RequestQueue对象。

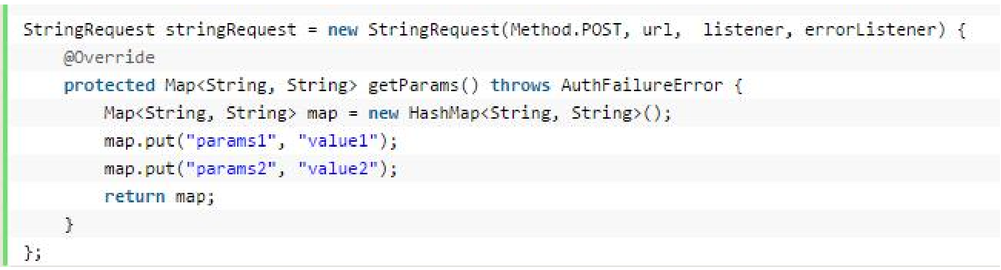
RequestQueue queue=volley.newRequestQueue(this);

2，创建一个StringRequest对象。

可指定请求方式。

post请求参数设置：

Volley会尝试调用StringRequest的父类——Request中的 getParams()方法来获取 POST参数



```
StringRequest stringRequest = new StringRequest(Method.POST, url,  listener, errorListener) {
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> map = new HashMap<String, String>();
        map.put("params1", "value1");
        map.put("params2", "value2");
        return map;
    }
};
```

3，将StringRequest对象添加到RequestQueue里面。

volley：

1，网络请求自动调度。

2，多个并发的网络请求。

3，Transparent disk and memory response caching with standard HTTP cache coherence.（透明的磁盘和内存响应缓存与标准的HTTP缓存一致性。）

4，支持请求优先级。

5，可以取消请求，设置取消请求范围。

6，调试和跟踪。

string，images，json

使用方式：

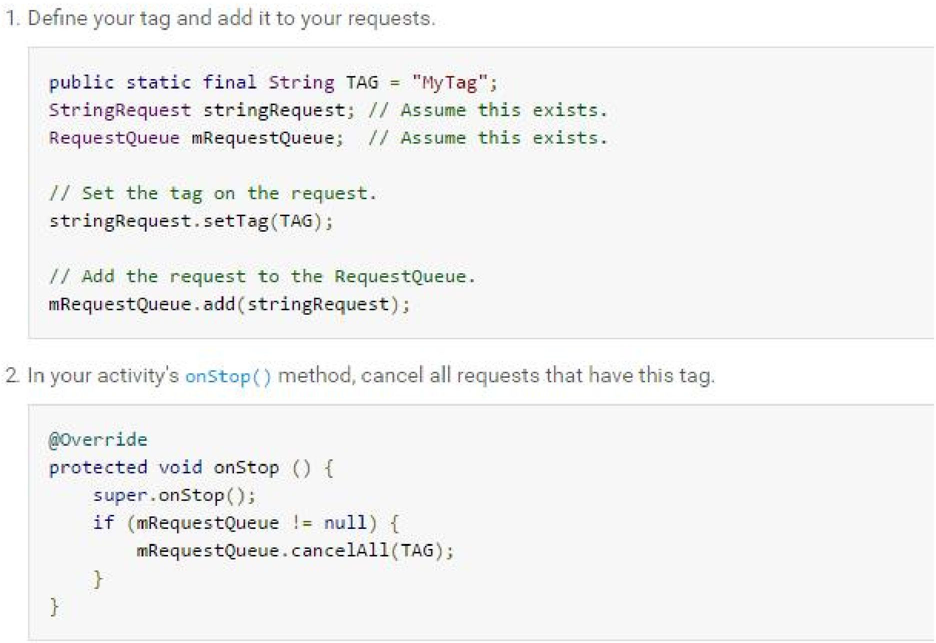
1，克隆到本地。

```
git clone https://android.googlesource.com/platform/frameworks/volley
```

2，Import the downloaded source into your app project as an Android library module as described in Create an Android Library.

运行在主线程上，you can freely modify UI controls directly from your response handler。

取消请求：



1. Define your tag and add it to your requests.

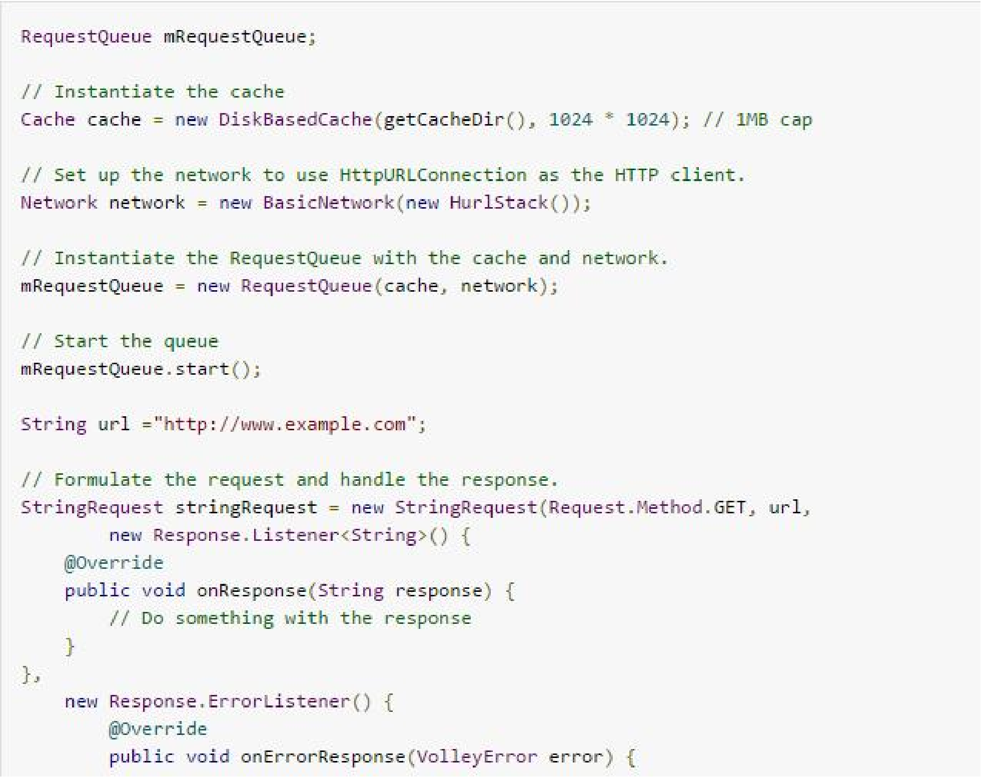
```
public static final String TAG = "MyTag";
StringRequest stringRequest; // Assume this exists.
RequestQueue mRequestQueue;  // Assume this exists.

// Set the tag on the request.
stringRequest.setTag(TAG);

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

2. In your activity's `onStop()` method, cancel all requests that have this tag.

```
@Override
protected void onStop () {
    super.onStop();
    if (mRequestQueue != null) {
        mRequestQueue.cancelAll(TAG);
    }
}
```

自定义请求：

定义缓存和网络



```
RequestQueue mRequestQueue;

// Instantiate the cache
Cache cache = new DiskBasedCache(getCacheDir(), 1024 * 1024); // 1MB cap

// Set up the network to use HttpURLConnection as the HTTP client.
Network network = new BasicNetwork(new HurlStack());

// Instantiate the RequestQueue with the cache and network.
mRequestQueue = new RequestQueue(cache, network);

// Start the queue
mRequestQueue.start();

String url ="http://www.example.com";

// Formulate the request and handle the response.
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        // Do something with the response
    }
},
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
```

图片请求：

1，ImageRequest

2，ImageLoader

3，NetWorkImageView

json数据请求