

## 使用 Volley 访问图片资源

### 知识解析

如果访问的资源是图片资源,则可以使用 `ImageRequest` 来访问网络。它的构造器如下:

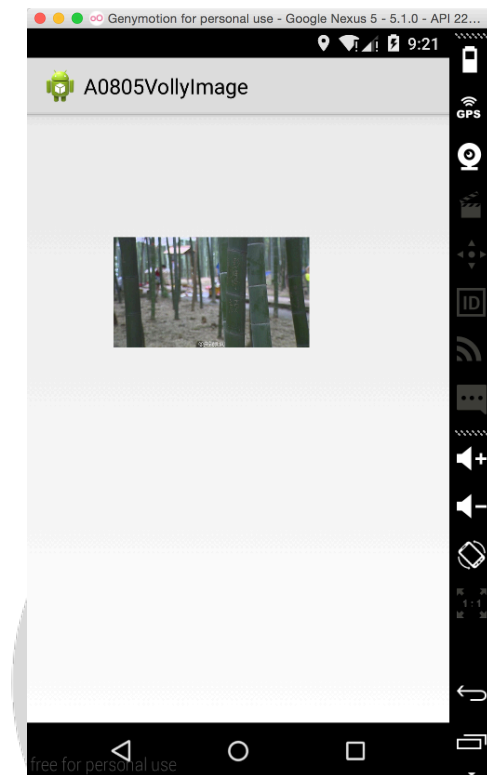
`ImageRequest(String url, Response.Listener<Bitmap> listener, int maxWidth, int maxHeight,`

`ScaleType scaleType, Config decodeConfig, Response.ErrorListener errorListener)`

第 3、4 个参数分别用于指定允许图片最大的宽度和高度,如果指定的网络图片的宽度或高度大于这里的最大值,则会对图片进行压缩,指定成 0 的话就表示不管图片有多大,都不会进行压缩。第 5 个参数是 `ImageView` 的 `ScaleType`。第 6 个参数用于指定图片的颜色属性,`Bitmap.Config` 下的几个常量都可以在这里使用。例如:

```
ImageRequest imgRequest = new ImageRequest(url,
    new Response.Listener<Bitmap>() {
        ...
    }, 0, 0, ImageView.ScaleType.FIT_XY,
    Bitmap.Config.ARGB_8888, new Response.ErrorListener()
    {...});
```

## 功能演示



## 操作实践

第一步，创建 `ImageView` 和 `ImageRquest` 对象

```
ImageView mImageView;
```

```
ImageRequest request;
```

第二步，实例化 `RequestQueue` 对象，并开始执行网络请求。

```
RequestQueue queue = Volley.newRequestQueue(this);  
String url =  
"http://file.yynet.com/2/1508/29/10341799-500.jpg";  
request = new ImageRequest(url, new  
Response.Listener<Bitmap>() {  
    @Override  
    public void onResponse(Bitmap response) {
```

```
// TODO Auto-generated method stub
mImageView.setImageBitmap(response);
}
}, 0, 0, ImageView.ScaleType.CENTER,
Bitmap.Config.ARGB_8888, new Response.ErrorListener() {

    @Override
    public void onErrorResponse (VolleyError error)
    {

        // TODO Auto-generated method stub
    }
});
queue.add(request);
```

## 职业素质

平时在开发Android应用的时候不可避免地都需要用到网络技术,而多数情况下应用程序都会使用HTTP协议来发送和接收网络数据。Android系统中主要提供了两种方式进行HTTP通信,HttpURLConnection和HttpClient,几乎在任何项目的代码中我们都能看到这两个类的身影,使用率非常高。

不过HttpURLConnection和HttpClient的用法还是稍微有些复杂的,如果不进行适当封装的话,很容易就会写出不少重复代码。于是乎,一些Android网络通信框架也就应运而生,比如说AsyncHttpClient,它把HTTP所有的通信细节全部封装在了内部,我们只需要简单调用几行代码就可以完成通信操作了。