

Android'de Volley Kütüphanesi İle Başağrsız Network İşlemleri

Mustafa Musa Ülker

Fatih Projesi Mobil Ekip Lideri

musaulker@gmail.com

Yasal Uyarı

Bu sunumda yer alan konular bilgilendirme amaçlı hazırlanmış olup tamamen kendi düşüncelerimi ifade etmekte, çalıştığım kurum adına bir görüş içermemektedir.

Ajanda

- Ben Kimim?
- Volley Kütüphanesi Nedir?
- Ne gibi avantajlar sunar?
- Kurulumu
- Kullanım Örnekleri

Mustafa Musa Ülker

- Bilgisayar Mühendisi
- e-Devlet Kapısı Projesi
- Fatih Projesi
- Mobil Uygulamalar
- Satranç oyuncusu :)

Volley Kütüphanesi

- Google Play ekibi tarafından geliştirilmiş **açık kaynak kodlu** bir Android network işlemleri kütüphanesidir.
- AOSP içerisinde yer almaktadır.

Özellikleri

- Kolay kullanım
- Hızlı network işlemleri (Paralel işleme)
- Transparan Disk ve Memory Cacheleme
- Network İşlemlerinin İptali
- Request Tracing Yapısı

Neden Volley?



CM101
F U N

cmymz.com

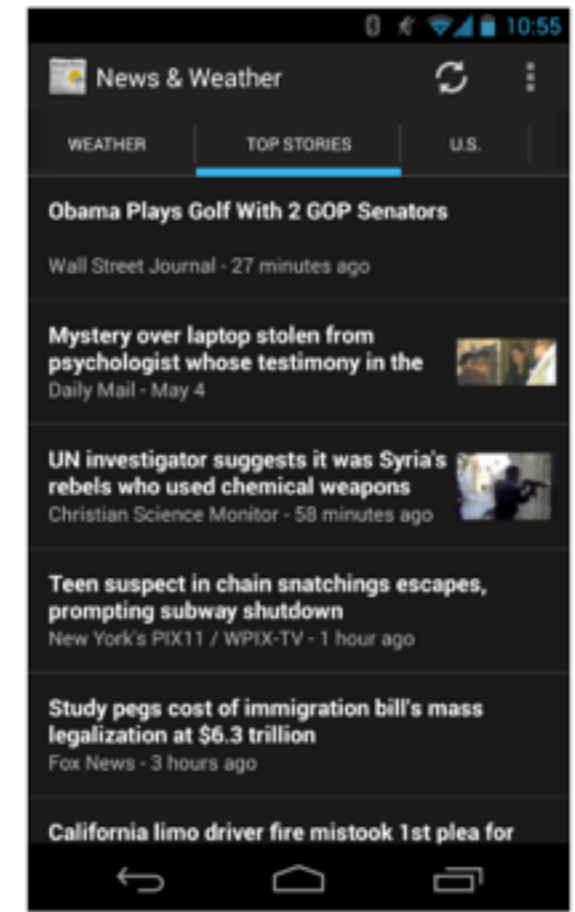
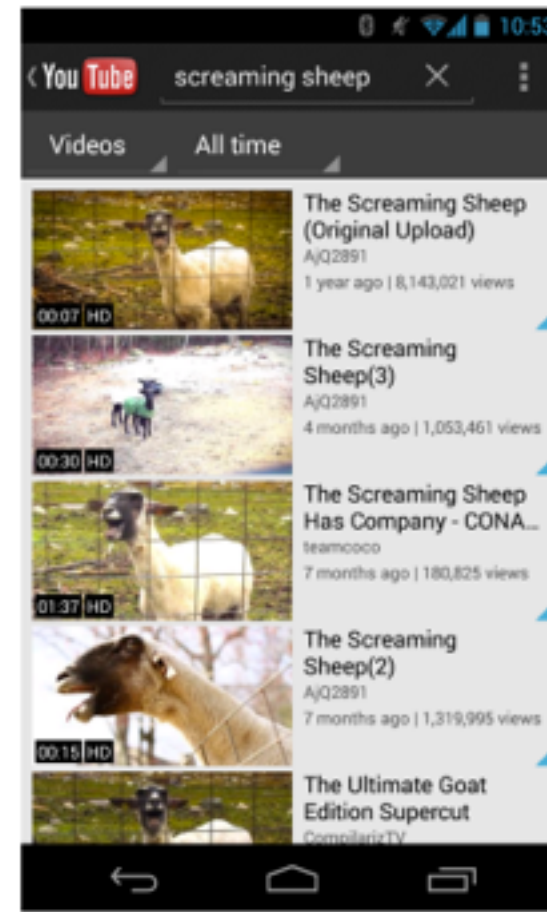
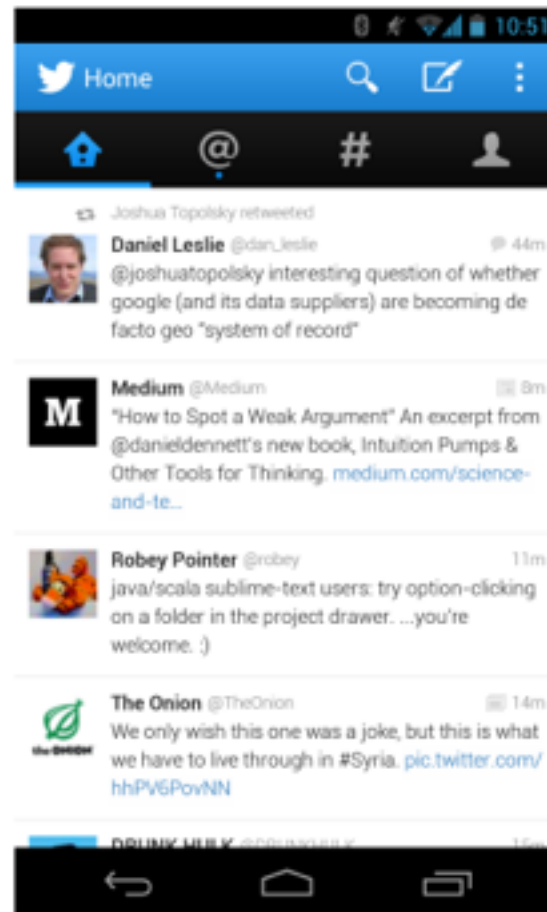
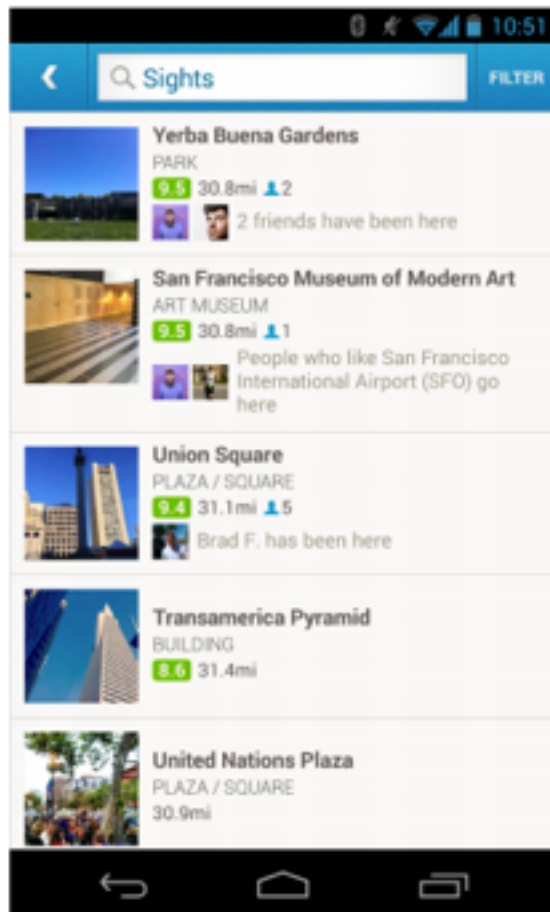
facebook.com/cmymz

twitter.com/cmymz

youtube.com/cmymz

İhtiyaçlar

- Bu uygulamaların ortak özelliği nedir?
- Network İşlemleri + UI Population + Detay ekranları



Geleneksel Problemler - 1

- **Problem:** Network requestleri sırayla çalıştırılır. FIFO
- **Çözüm 1:** AsyncTask.THREAD_POOL_EXECUTOR
- **Çözüm 2:** AsyncTask.executeOnExecutor(Executor exec, Params... params)

Kaynakça: <http://developer.android.com/reference/android/os/AsyncTask.html>

Geleneksel Problemler - 1

- **Problem:** Network requestleri sırayla çalıştırılır. FIFO

Volley Çözümü

- Otomatik olarak ThreadPool içerisinde network isteklerini paralel olarak gerçekleştirir. (4 Thread)
- İstek Önceliklendirme (Yüksek, Orta veya Düşük)
- Önce veriyi çek (yüksek öncelik) sonra diğer resmi yüklemeye geç (düşük öncelik) gibi

Geleneksel Problemler - 2

- **Problem:** Ekranın dönmesi = bütün requestlerin tekrarlanması
- **Çözüm:** HashMap, LruCache vs. kullanarak çözülebilir.

Kaynakça: <http://developer.android.com/reference/android/util/LruCache.html>

Geleneksel Problemler - 2

- **Problem:** Ekranın dönmesi = bütün requestlerin tekrarlanması

Volley Çözümü

- Transparan Disk/Memory Cache sunuyor.
- Sorgular sırasında cache var mı yok mu kontrol etmeye gerek yok. Arkaplanda kendisi hallediyor.
- Hızlı!

Geleneksel Problemler - 3

- **Problem:** Birden fazla AsyncTask Yönetimi (resimler)
- **Çözüm:** AsyncTask'lerin takip edilmesi ve işi biten ImageView'ların recycle() edilmesi

Kaynakça: <http://developer.android.com/training/displaying-bitmaps/manage-memory.html>

Geleneksel Problemler - 3

- **Problem:** Birden fazla AsyncTask Yönetimi (resimler)

Volley Çözümü

- Request cancelling
- `com.android.volley.NetworkImageView`

Ne zaman Volley?

- Arayüzün network üzerinden çekilerek oluşturulan uygulamalar için **mükemmel!**
- Arkaplanda küçük network işlemleri için **güzel!**
- Büyük veriler (dosya indirme gibi) için **dikkat!**

Kurulum

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

```
$ cd volley
```

```
$ android update project -p .
```

```
$ ant jar
```

```
$ cp bin/volley.jar myproject/libs
```


Temel Kullanımı

```
RequestQueue myVolleyQueue = volley.newRequestQueue(Context);
```

- **RequestQueue** nesnesinden değişkenimizi oluşturuyoruz.

```
(JsonObjectRequest | JSONArrayRequest | StringRequest) request;  
...  
myVolleyQueue.add(request);
```

- **Request** nesnelerimizi tanımlayıp, kuyruğa ekliyoruz.

Geleneksel Yöntem

- HttpClient + HttpGet + HttpResponse + InputStream

```
public JSONArray getContentList() {
    DefaultHttpClient httpClient = new DefaultHttpClient(new BasicHttpParams());
    HttpGet httpget = new HttpGet("http://api.mywebsite.com/contentlist");
    httpget.setHeader("Content-type", "application/json");

    InputStream inputStream = null;
    JSONArray result = null;
    try {
        HttpResponse response = httpClient.execute(httpget);
        HttpEntity entity = response.getEntity();

        inputStream = entity.getContent();
        BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream, "UTF-8"),
8);
        StringBuilder sb = new StringBuilder();

        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        result = new JSONArray(sb.toString());
    } catch (Exception e) {
        // continue
    } finally {
        try {
            if (inputStream != null)
                inputStream.close();
        } catch (Exception squish) {
        }
    }
    return result;
}
```


Volley Yaklaşımı

- RequestQueue + JsonRequest

```
public void getContentList() {
    RequestQueue myVolleyQueue = Volley.newRequestQueue(this);
    JsonRequest req = new JsonRequest("http://
api.mywebsite.com/contentlist", new Response.Listener<JSONArray>()
{
    @Override
    public void onResponse(JSONArray response) {
        // we have response here
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

    }
});
myVolleyQueue.add(req);
}
```


Resim Yükleme Senaryoları

- `ImageRequest`: URL -> Bitmap
- `ImageDownloader`: Cacheleme yapar
- `NetworkImageView`: Yüğü üzerinden alır :)

ImageRequest

- ImageRequest: URL -> Bitmap

```
ImageRequest myImageRequest = new ImageRequest(<URL>, new
Response.Listener<Bitmap>() {
    @Override
    public void onResponse(Bitmap response) {
        mImageView.setImageBitmap(response);
    }
}, 0, 0, Bitmap.Config.ARGB_8888, new Response.ErrorListener()
{
    @Override
    public void onErrorResponse(VolleyError error) {
        mImageView.setImageResource(R.drawable.error);
    }
});
myVolleyQueue.add(myImageRequest);
```


ImageDownloader

- ImageDownloader: Cacheleme

```
int memoryClass = ((ActivityManager)
context.getSystemService(Context.ACTIVITY_SERVICE))
    .getMemoryClass();
int cacheSize = 1024 * 1024 * memoryClass / 8;

ImageLoader myImageLoader = new ImageLoader(myVolleyQueue, new
BitmapCache(cacheSize));

myImageLoader(<URL>,
    ImageLoader.getImageListener(myImageView, // Normal ImageView
        R.drawable.flickr,
        android.R.drawable.ic_dialog_alert),
        50, 50);
```


NetworkImageView

- NetworkImageView: ImageView yerine

```
myNetworkImageView.setImageUrl(url, myImageLoader);
```

- GC temizleme konusunda daha hassas
- ImageView için;
 - Yeni bir request gelirse
 - veya ekran değişirse

bütün veriyi temizliyor

Request Cancellation

- `setTag()` + `cancelAll()`

```
RequestQueue myVolleyQueue = volley.newRequestQueue(this);  
StringRequest myRequest = new StringRequest(...)
```

```
...  
...
```

```
// Set Request Tag  
myRequest.setTag("content");
```

```
// Add request to Queue  
myVolleyQueue.add(myRequest);
```

```
// Cancel all requests with "content" tag!  
myVolleyQueue.cancelAll("content");
```


Request Önceliklendirme

- ImageRequests - Low
- Diğerleri - Normal

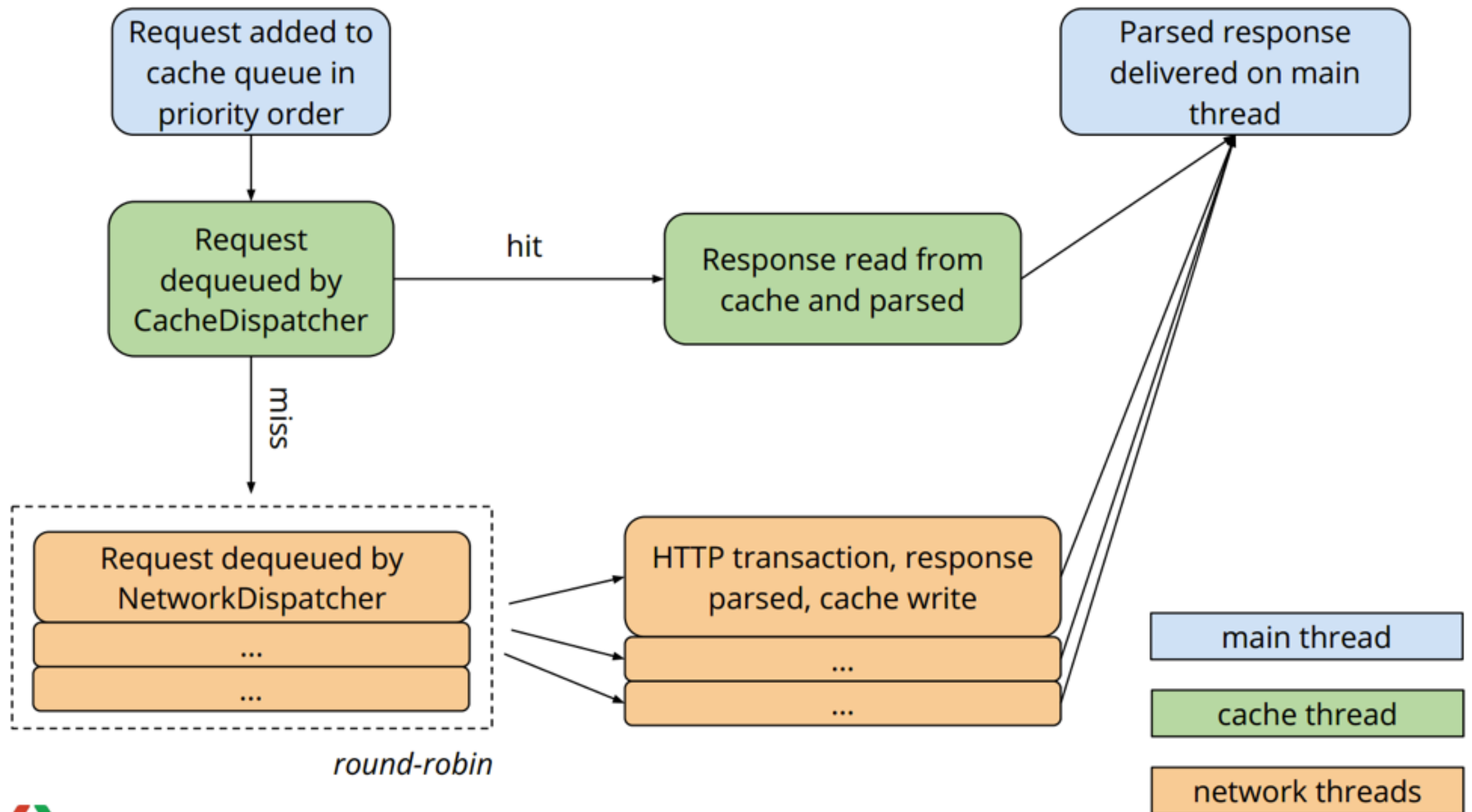
```
RequestQueue myVolleyQueue = volley.newRequestQueue(this);  
StringRequest myRequest = new StringRequest(...)
```

```
...  
...
```

```
// Set Request Priority HIGH  
myRequest.setPriority(Priority.HIGH);
```

```
// Add request to Queue  
myVolleyQueue.add(myRequest);
```


Volley Mimarisi



Debug

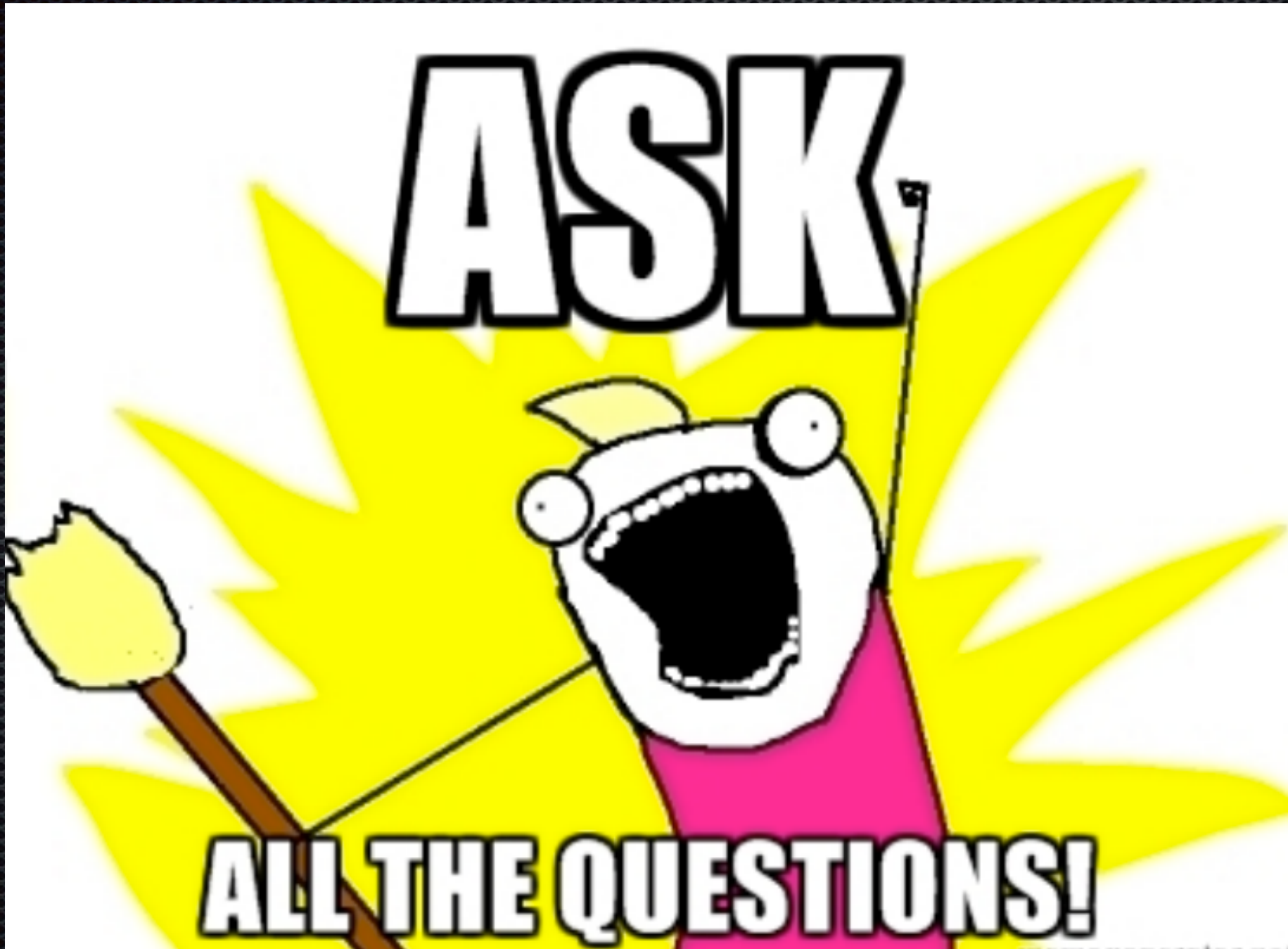
```
$ adb shell setprop log.tag.volley VERBOSE
```

```
com.musaulker.devf... Volley [1] MarkerLog.finish: (1877 ms) [ ] https://picasaweb.google.com/da ↵  
ta/feed/api/all?q=kitten&max-results=20&thumbsize=160&alt=json&star ↵  
t-index=41 0x16731b0b NORMAL 25  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+0 ) [ 1] add-to-queue  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+1 ) [12021] cache-queue-take  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+0 ) [12021] cache-miss  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+0 ) [12024] network-queue-take  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+1754) [12024] network-http-complete  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+108 ) [12024] network-parse-complete  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+13 ) [12024] network-cache-written  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+0 ) [12024] post-response  
com.musaulker.devf... Volley [1] MarkerLog.finish: (+1 ) [ 1] done
```


Kaynakça

- Google I/O 2013 - Ficus Kirkpatrick Session
- Ognyan Bankov Blog
- API Dokümantasyonu: <http://files.evancharlton.com/volley-docs/>
- Stackoverflow

Sorular?



Teşekkürler

- Katıldığınız için teşekkür ediyorum :)
- Sunum ve kodlar; http://github.com/musaulker/devfesttr_2013_android_volley

İletişim

Mustafa Musa Ülker

musaulker@gmail.com

<http://twitter.com/musaulker>