



FREE eBook

LEARNING android-volley

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
volley

Table of Contents

About.....	1
Chapter 1: Getting started with android-volley.....	2
Remarks.....	2
Examples.....	2
Installation And Setup.....	2
Credits.....	6

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [android-volley](#)

It is an unofficial and free android-volley ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-volley.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with android-volley

Remarks

This section provides an overview of what android-volley is, and why a developer might want to use it.

It should also mention any large subjects within android-volley, and link out to the related topics. Since the Documentation for android-volley is new, you may need to create initial versions of those related topics.

Examples

Installation And Setup

Installation

Volley JCenter Gradle Import

```
//in your project's app level build.gradle
compile 'com.android.volley:volley:1.0.0'
```

Create a subclass of Application

```
public class ApplicationController extends Application {

    public static final String TAG = ApplicationController.class
        .getSimpleName();

    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;

    private static ApplicationController mInstance;

    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }

    public static synchronized ApplicationController getInstance() {
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(getApplicationContext());
        }

        return mRequestQueue;
    }
}
```

```

public ImageLoader getImageLoader() {
    getRequestQueue();
    if (mImageLoader == null) {
        mImageLoader = new ImageLoader(this.mRequestQueue,
            new LruBitmapCache());
    }
    return this.mImageLoader;
}

public <T> void addToRequestQueue(Request<T> req, String tag) {
    // set the default tag if tag is empty
    req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
    getRequestQueue().add(req);
}

public <T> void addToRequestQueue(Request<T> req) {
    req.setTag(TAG);
    getRequestQueue().add(req);
}

public void cancelPendingRequests(Object tag) {
    if (mRequestQueue != null) {
        mRequestQueue.cancelAll(tag);
    }
}
}
}

```

Create a StringRequest

```

public class StringRequestActivity extends Activity {

    private String TAG = StringRequestActivity.class.getSimpleName();
    private Button btnStringReq;
    private TextView msgResponse;

    // This tag will be used to cancel the request
    private String tag_string_req = "string_req";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_string);

        btnStringReq = (Button) findViewById(R.id.btnStringReq);
        msgResponse = (TextView) findViewById(R.id.msgResponse);

        btnStringReq.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                makeStringReq();
            }
        });
    }

    /**
     * Making json object request

```

```

* */
private void makeStringReq() {

    StringRequest strReq = new StringRequest(Method.GET,
        "http://www.myurl.com", new Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                Log.d(TAG, response.toString());
                msgResponse.setText(response.toString());
            }
        }, new Response.ErrorListener() {

            @Override
            public void onErrorResponse(VolleyError error) {
                VolleyLog.d(TAG, "Error: " + error.getMessage());
            }
        }
    });

    // Adding request to request queue
    ApplicationController.getInstance().addToRequestQueue(strReq, tag_string_req);
}
}

```

Singleton RequestQueue Setup

Generally it's recommended that you use a single RequestQueue throughout your Application. So, you want to have one NetworkRequestManager singleton that contains your Volley RequestQueue. A simple implementation would be:

```

public class NetworkRequestManager {

    private static final String TAG = NetworkRequestManager.class.getName();

    private static NetworkRequestManager mManager;

    private RequestQueue requestQueue;

    private NetworkRequestManager(@NonNull final Context context) {
        initQueue(context);
    }

    /**
     * @return A NetworkRequestManager with an initialized RequestQueue
     */
    public static synchronized NetworkRequestManager getInstance(@NonNull final Context context) {
        if(mManager == null){
            mManager = new NetworkRequestManager(context);
        }
        return mManager;
    }

    /**
     * Initialize your request queue. This uses the default Volley

```

```

* setup.
*
* @param context
*/
private void initQueue(Context context) {
    if (requestQueue == null) {
        requestQueue = Volley.newRequestQueue(context.getApplicationContext());
    }
}

/**
 * @Return our initialized RequestQueue
 */
public RequestQueue getRequestQueue() {
    return requestQueue;
}

/**
 * Cancels all requests for the given object tag
 *
 * @param tag
 */
@Override
public void cancelAllForTag(Object tag) {
    getRequestQueue().cancelAll(tag);
}

/**
 *
 * A convenience method for adding requests to the queue that are associated with a tag for
cancellation
 *
 * @param request
 * @param tag
 */
@Override
public void addRequest(Request<?> request, Object tag) {
    request.setTag(tag);
    getRequestQueue().add(request);
}

```

Read [Getting started with android-volley](https://riptutorial.com/android-volley/topic/4794/getting-started-with-android-volley) online: <https://riptutorial.com/android-volley/topic/4794/getting-started-with-android-volley>

Credits

S. No	Chapters	Contributors
1	Getting started with android-volley	Adriana Carelli , Community , spuente , Submersed