



EBook Gratis

APRENDIZAJE android-volley

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
volley

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con android-volley.....	2
Observaciones.....	2
Examples.....	2
Instalación y configuración.....	2
Creditos.....	6

Acerca de

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

Capítulo 1: Empezando con android-volley

Observaciones

Esta sección proporciona una descripción general de qué es android-volley y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante dentro de android-volley, y vincular a los temas relacionados. Dado que la Documentación para android-volley es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación y configuración

Instalación

Volley JCenter Gradle Import

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

Crear una subclase de aplicación

```
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);
    }
}
}

```

Crear un 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);

}
}

```

Configuración de RequestQueue Singleton

En general, se recomienda que utilice un único RequestQueue en toda su aplicación. Por lo tanto, desea tener un singleton NetworkRequestManager que contenga su Volley RequestQueue. Una implementación simple sería:

```

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);
}

```

Lea Empezando con android-volley en línea: <https://riptutorial.com/es/android-volley/topic/4794/empezando-con-android-volley>

Creditos

S. No	Capítulos	Contributors
1	Empezando con android-volley	Adriana Carelli , Community , spuente , Submersed