Android Volley Library Tutorial - Fetch JSON from Fake API

Introduction to Volley

Volley is an HTTP library developed by Google that makes networking for Android apps easier and faster. It handles the complexities of network requests, caching, and threading automatically.

Key Features:

- Automatic scheduling of network requests
- Multiple concurrent network connections
- Transparent disk and memory caching
- Request prioritization
- Cancellation request API
- Easy customization

Setup

Step 1: Add Volley Dependency

Add this to your `build.gradle (Module: app)` file:

```gradle
dependencies {
    implementation 'com.android.volley:volley:1.2.1'
}
```

Step 2: Add Internet Permission

Add this to your `AndroidManifest.xml`:

```xml
<uses-permission android:name="android.permission.INTERNET" />
```

Fake API - JSONPlaceholder

We'll use **JSONPlaceholder** (https://jsonplaceholder.typicode.com/) - a free fake REST API for testing.

Available Endpoints:

- `/posts` - 100 posts
- `/comments` - 500 comments

- `/users` - 10 users
- `/albums` - 100 albums
- `/photos` - 5000 photos
- `/todos` - 200 todos

Sample JSON Response (Single Post):

```json
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident",
  "body": "quia et suscipit\nsuscipit recusandae..."
}
```

Example 1: Simple GET Request - Fetch All Posts

Model Class: Post.java

```java
```

```java
public class Post {
    private int userId;
    private int id;
    private String title;
    private String body;

    // Constructor
    public Post(int userId, int id, String title, String body) {
        this.userId = userId;
        this.id = id;
        this.title = title;
        this.body = body;
    }

    // Getters
    public int getUserId() { return userId; }
    public int getId() { return id; }
    public String getTitle() { return title; }
    public String getBody() { return body; }

    // Setters
    public void setUserId(int userId) { this.userId = userId; }
    public void setId(int id) { this.id = id; }
    public void setTitle(String title) { this.title = title; }
    public void setBody(String body) { this.body = body; }
}
```

Activity: MainActivity.java

```java
```

```
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private TextView textViewResult;
    private RequestQueue requestQueue;
    private List<Post> postList;

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

        textViewResult = findViewById(R.id.textViewResult);
        postList = new ArrayList<>();

        // Initialize RequestQueue
        requestQueue = Volley.newRequestQueue(this);

        // Fetch posts
        fetchPosts();
    }

    private void fetchPosts() {
        String url = "https://jsonplaceholder.typicode.com/posts";

        JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(
            Request.Method.GET,
            url,
            null,
            new Response.Listener<JSONArray>() {
                @Override
```

```
            public void onResponse(JSONArray response) {
                try {
                    // Parse JSON Array
                    for (int i = 0; i < response.length(); i++) {
                        JSONObject postObject = response.getJSONObject(i);

                        int userId = postObject.getInt("userId");
                        int id = postObject.getInt("id");
                        String title = postObject.getString("title");
                        String body = postObject.getString("body");

                        Post post = new Post(userId, id, title, body);
                        postList.add(post);
                    }

                    // Display first 5 posts
                    displayPosts();

                } catch (JSONException e) {
                    e.printStackTrace();
                    Toast.makeText(MainActivity.this,
                        "Error parsing JSON", Toast.LENGTH_SHORT).show();
                }
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(MainActivity.this,
                    "Error: " + error.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    );

    // Add request to queue
    requestQueue.add(jsonArrayRequest);
}

private void displayPosts() {
    StringBuilder result = new StringBuilder();

    // Display first 5 posts only
    int limit = Math.min(5, postList.size());
    for (int i = 0; i < limit; i++) {
        Post post = postList.get(i);
        result.append("ID: ").append(post.getId()).append("\n");
```

```
            result.append("Title: ").append(post.getTitle()).append("\n");
            result.append("Body: ").append(post.getBody()).append("\n\n");
        }

        textViewResult.setText(result.toString());
    }
}
```

Layout: activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Volley API Example"
            android:textSize="24sp"
            android:textStyle="bold"
            android:gravity="center"
            android:layout_marginBottom="20dp"/>

        <TextView
            android:id="@+id/textViewResult"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Loading..."
            android:textSize="14sp"
            android:lineSpacingExtra="4dp"/>

    </LinearLayout>
</ScrollView>
```

Example 2: GET Single Post by ID

```java
```

```
private void fetchSinglePost(int postId) {
    String url = "https://jsonplaceholder.typicode.com/posts/" + postId;

    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(
        Request.Method.GET,
        url,
        null,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    int userId = response.getInt("userId");
                    int id = response.getInt("id");
                    String title = response.getString("title");
                    String body = response.getString("body");

                    String result = "ID: " + id + "\n" +
                            "User ID: " + userId + "\n" +
                            "Title: " + title + "\n" +
                            "Body: " + body;

                    textViewResult.setText(result);

                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(MainActivity.this,
                    "Error: " + error.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    );

    requestQueue.add(jsonObjectRequest);
}
```

Example 3: POST Request - Create New Post

java

```
private void createPost() {
    String url = "https://jsonplaceholder.typicode.com/posts";

    // Create JSON Object to send
    JSONObject postData = new JSONObject();
    try {
        postData.put("title", "My New Post");
        postData.put("body", "This is the body of my post");
        postData.put("userId", 1);
    } catch (JSONException e) {
        e.printStackTrace();
    }

    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(
        Request.Method.POST,
        url,
        postData,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    String result = "Post Created!\n\n" +
                                    "ID: " + response.getInt("id") + "\n" +
                                    "Title: " + response.getString("title") + "\n" +
                                    "Body: " + response.getString("body") + "\n" +
                                    "User ID: " + response.getInt("userId");

                    textViewResult.setText(result);
                    Toast.makeText(MainActivity.this,
                        "Post created successfully!",
                        Toast.LENGTH_SHORT).show();

                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(MainActivity.this,
                    "Error: " + error.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    );
```

```
    requestQueue.add(jsonObjectRequest);
}
```

Example 4: RecyclerView with Volley

Model: User.java

java

```
public class User {
    private int id;
    private String name;
    private String email;
    private String phone;

    public User(int id, String name, String email, String phone) {
        this.id = id;
        this.name = name;
        this.email = email;
        this.phone = phone;
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public String getEmail() { return email; }
    public String getPhone() { return phone; }
}
```

Adapter: UserAdapter.java

java

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class UserAdapter extends RecyclerView.Adapter<UserAdapter.UserViewHolder> {

    private List<User> userList;

    public UserAdapter(List<User> userList) {
        this.userList = userList;
    }

    @NonNull
    @Override
    public UserViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.item_user, parent, false);
        return new UserViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull UserViewHolder holder, int position) {
        User user = userList.get(position);
        holder.textViewName.setText(user.getName());
        holder.textViewEmail.setText(user.getEmail());
        holder.textViewPhone.setText(user.getPhone());
    }

    @Override
    public int getItemCount() {
        return userList.size();
    }

    public static class UserViewHolder extends RecyclerView.ViewHolder {
        TextView textViewName, textViewEmail, textViewPhone;

        public UserViewHolder(@NonNull View itemView) {
            super(itemView);
            textViewName = itemView.findViewById(R.id.textViewName);
            textViewEmail = itemView.findViewById(R.id.textViewEmail);
            textViewPhone = itemView.findViewById(R.id.textViewPhone);
        }
```

```
    }
}
```

Layout: item_user.xml

```xml
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <TextView
            android:id="@+id/textViewName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Name"
            android:textSize="18sp"
            android:textStyle="bold"
            android:textColor="#000000"/>

        <TextView
            android:id="@+id/textViewEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Email"
            android:textSize="14sp"
            android:layout_marginTop="4dp"/>

        <TextView
            android:id="@+id/textViewPhone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Phone"
            android:textSize="14sp"
            android:layout_marginTop="4dp"/>

    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Activity: UsersActivity.java

java

```
import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

public class UsersActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private UserAdapter adapter;
    private List<User> userList;
    private RequestQueue requestQueue;

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

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        userList = new ArrayList<>();
        adapter = new UserAdapter(userList);
        recyclerView.setAdapter(adapter);

        requestQueue = Volley.newRequestQueue(this);

        fetchUsers();
    }

    private void fetchUsers() {
        String url = "https://jsonplaceholder.typicode.com/users";

        JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(
            Request.Method.GET,
```

```
            url,
            null,
            new Response.Listener<JSONArray>() {
                @Override
                public void onResponse(JSONArray response) {
                    try {
                        for (int i = 0; i < response.length(); i++) {
                            JSONObject userObject = response.getJSONObject(i);

                            int id = userObject.getInt("id");
                            String name = userObject.getString("name");
                            String email = userObject.getString("email");
                            String phone = userObject.getString("phone");

                            User user = new User(id, name, email, phone);
                            userList.add(user);
                        }

                        adapter.notifyDataSetChanged();

                    } catch (JSONException e) {
                        e.printStackTrace();
                        Toast.makeText(UsersActivity.this,
                            "Error parsing data", Toast.LENGTH_SHORT).show();
                    }
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(UsersActivity.this,
                        "Error: " + error.getMessage(),
                        Toast.LENGTH_SHORT).show();
                }
            }
        );

        requestQueue.add(jsonArrayRequest);
    }
}
```

Best Practices

1. Create a Singleton RequestQueue

java

```java
public class VolleySingleton {
    private static VolleySingleton instance;
    private RequestQueue requestQueue;
    private static Context context;

    private VolleySingleton(Context ctx) {
        context = ctx;
        requestQueue = getRequestQueue();
    }

    public static synchronized VolleySingleton getInstance(Context context) {
        if (instance == null) {
            instance = new VolleySingleton(context);
        }
        return instance;
    }

    public RequestQueue getRequestQueue() {
        if (requestQueue == null) {
            requestQueue = Volley.newRequestQueue(context.getApplicationContext());
        }
        return requestQueue;
    }

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

2. Cancel Requests

```java
// Tag requests
jsonArrayRequest.setTag("MY_TAG");

// Cancel when activity is destroyed
@Override
protected void onStop() {
    super.onStop();
    if (requestQueue != null) {
        requestQueue.cancelAll("MY_TAG");
    }
}
```

3. Handle Timeouts

```java
jsonArrayRequest.setRetryPolicy(new DefaultRetryPolicy(
    30000, // 30 seconds timeout
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
));
```

Common Volley Request Types

1. **StringRequest** - Returns raw string response
2. **JsonObjectRequest** - Returns JSONObject
3. **JsonArrayRequest** - Returns JSONArray
4. **ImageRequest** - For loading images

Assignment Ideas for Students

1. **Beginner**: Fetch and display all posts in a TextView
2. **Intermediate**: Create a RecyclerView showing all users with their details
3. **Advanced**: Create a full CRUD app (Create, Read, Update, Delete) for posts
4. **Challenge**: Implement search functionality to filter posts by title

---

Additional Resources

- Official Volley Documentation: https://developer.android.com/training/volley
- JSONPlaceholder API Docs: https://jsonplaceholder.typicode.com/
- Alternative Fake APIs:
  - DummyJSON: https://dummyjson.com/
  - ReqRes: https://reqres.in/

Common Errors and Solutions

Error 1: NetworkOnMainThreadException

Solution: Volley handles this automatically by running requests on background threads.

Error 2: No Internet Permission

Solution: Add `<uses-permission android:name="android.permission.INTERNET" />` to AndroidManifest.xml

Error 3: Cleartext Traffic Not Permitted (Android 9+)

Solution: Use HTTPS URLs (JSONPlaceholder supports both HTTP and HTTPS)

Summary

Volley simplifies Android networking by:

- Handling threading automatically
- Providing built-in caching
- Supporting request cancellation
- Offering easy JSON parsing

With JSONPlaceholder, students can practice API integration without needing a backend server!