OpenWeatherMap API with Volley

John P. Baugh, Ph.D.

Format of JSON returned from a simple API request

When you make a simple request using something like the following, but if you try it out, **USE YOUR OWN API KEY!!!!! (the part after &appid=……)**

```
package com.testname.weatherapphelp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;

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.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class MainActivity extends AppCompatActivity {

    private RequestQueue requestQueue;

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

        //instantiate the request queue
        requestQueue = Volley.newRequestQueue(this);


        //create object request
        JsonObjectRequest jsonObjectRequest =
                new JsonObjectRequest(
                        Request.Method.GET,    //the request method

"https://api.openweathermap.org/data/2.5/weather?q=London&appid=1c7965dd72e8c05e2f557704d77c998b",
                        null,
```

```
                        new Response.Listener<JSONObject>() {
                            @Override
                            public void onResponse(JSONObject response) {
                                //this prints the WHOLE string
                                //Log.i("JSON response", response.toString());

                                try {
                                    //get description of weather
                                    JSONArray weather = response.getJSONArray("weather");

                                    //since it's one day of weather,
                                    //there's one object in the array
                                    JSONObject currentWeather = weather.getJSONObject(0);

                                    int id = currentWeather.getInt("id");
                                    String mainWeather = currentWeather.getString("main");
                                    String description =
                                            currentWeather.getString("description");

                                    Log.i("JSON info", "ID: " + id);
                                    Log.i("JSON info", "main weather: " + mainWeather);
                                    Log.i("JSON info", "Description: " + description);
                                }
                                catch(JSONException ex) {
                                    Log.e("JSON Error", ex.getMessage());
                                }
                            }
                        },
                        new Response.ErrorListener(){

                            @Override
                            public void onErrorResponse(VolleyError error) {

                            }
                        }
                );//end of JSON object request

                requestQueue.add(jsonObjectRequest);

    }//end onCreate
}
```

You get returned JSON like the following (I've formatted it to look nice, but it's just returned as a regular String essentially)


```
{
	"coord": {
		"lon": -0.13,
		"lat": 51.51
	},
	"weather": [
		{
			"id": 802,
			"main": "Clouds",
			"description": "scattered clouds",
			"icon": "03n"
		}
	],
	"base": "stations",
	"main": {
		"temp": 287.31,
		"feels_like": 282.62,
		"temp_min": 283.71,
		"temp_max": 290.93,
		"pressure": 1026,
		"humidity": 62
	},
	"visibility": 10000,
	"wind": {
		"speed": 5.7,
		"deg": 100
	},
	"clouds": {
		"all": 41
	},
	"dt": 1586461451,
	"sys": {
		"type": 1,
		"id": 1414,
		"country": "GB",
		"sunrise": 1586409413,
		"sunset": 1586458000
	},
	"timezone": 3600,
	"id": 2643743,
	"name": "London",
	"cod": 200
}
```


Do you see how the entire JSON is enclosed in curly braces? { … }

That is the outermost JSONObject. The outermost object has the following subproperties:

- coord
- weather
- base
- main
- visibility
- wind
- clouds
- dt
- sys
- timezone
- id
- name
- cod

As examples of subproperties and how to interpret them, **coord** is an object, since you'll notice after the colon there is a set of curly braces, { … }

The **weather** subproperty is an **array** (a **JSONArray**), because instead of curly braces, it uses square brackets.

The **base** property is a **string** so that is fairly straightforward.

In the Java earlier, the method:

```
public void onResponse(JSONObject response) {
    //...
}
```

This method has a parameter automatically passed to it named response. **This JSONObject is the outermost received JSON object**. Therefore you can use the **response** object to access sub-properties.

Notice that I do so with **the weather object**:

```
"weather": [
            {
                  "id": 802,
                  "main": "Clouds",
                  "description": "scattered clouds",
                  "icon": "03n"
            }
      ],
```

Note that the weather array only has a **single object** (in { … }) inside of it. So that's at index 0. So **weather[0]** returns the object containing id, main, description, and icon.

I access the weather object and store its reference in a JSONArray object:

```
//get description of weather
JSONArray weather = response.getJSONArray("weather");

//since it's one day of weather,
//there's one object in the array
JSONObject currentWeather = weather.getJSONObject(0);
```

So in the above code, we get the **weather** property from the outermost object (response), and that weather property happens to be an array. THEN, we get the first (which in this case happens to be the only) object out of the weather array with **getJSONObject(0)**.

After we get that, then **currentWeather** contains:

```
{
      "id": 802,
      "main": "Clouds",
      "description": "scattered clouds",
      "icon": "03n"
}
```

The **currentWeather** object contains the four properties: id, main, description, and icon. The first (id) is an integer, and the other three are strings, so we can use getInt and getString on them.

And that's how that works! 😊