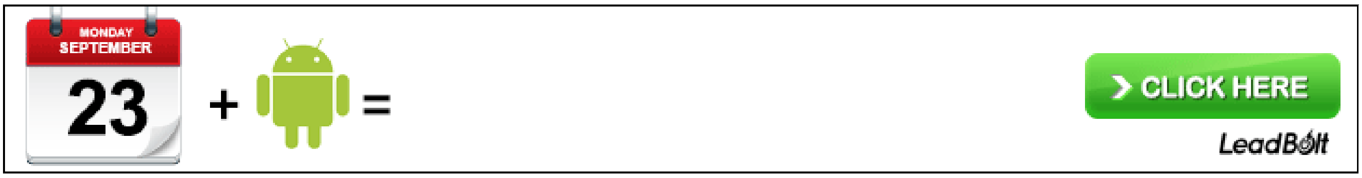


January 21, 2012 04:41 PM

Android JSON Parsing Tutorial

392 Comments Like 338 Tweet 42

JSON is the best alternative to XML for storing data in files. It is easy to parse and access data stored in JSON format. Previously i explained parsing XML and today i am going to discuess parsing JSON data with an example.

DOWNLOAD CODE

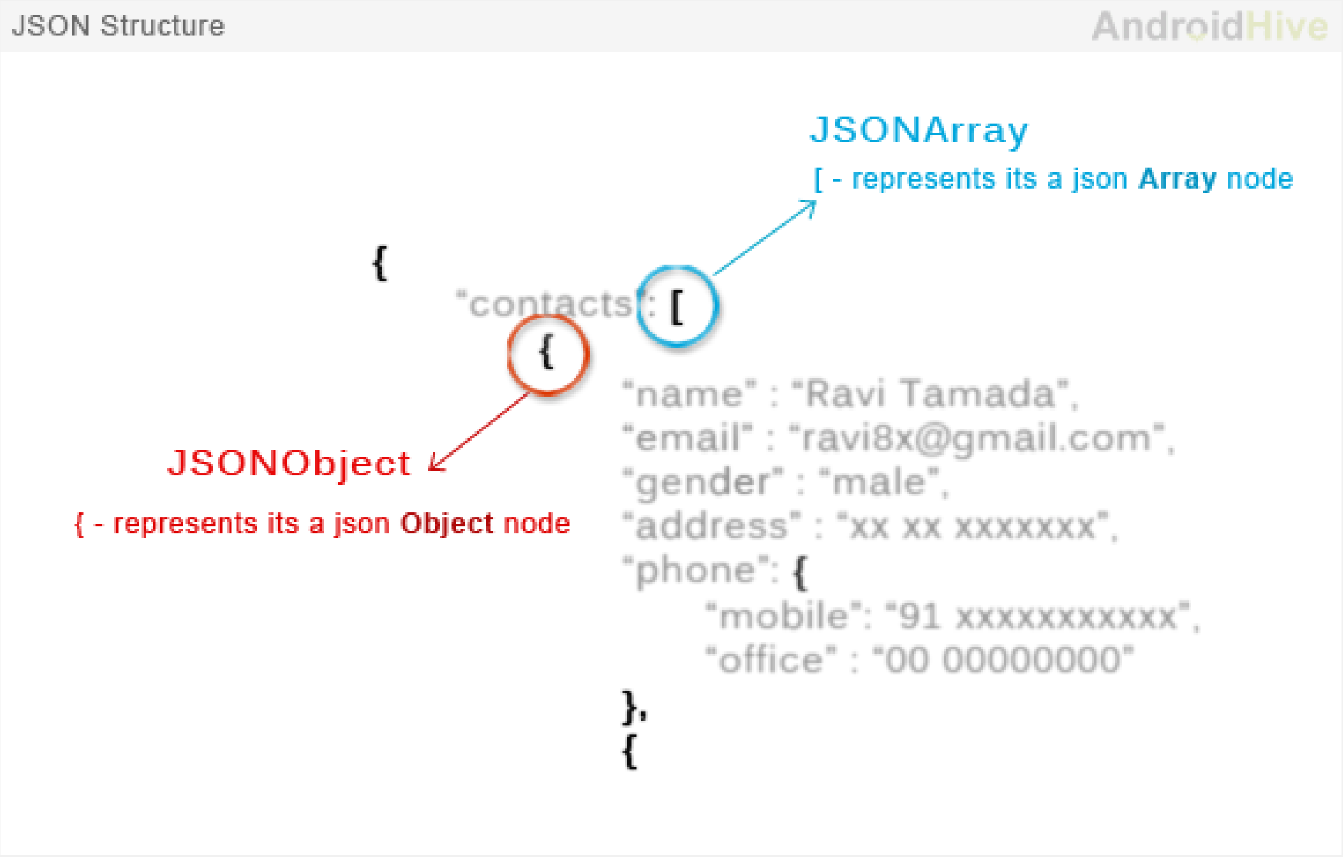
The JSON Structure

I am taking an example of following JSON which will give you list of contacts and each contact will have details like name, email, address, phone number ertc,. You can get this JSON data by accessing http://api.androidhive.info/contacts/

```
{
    "contacts": [
        {
                "id": "c200",
                "name": "Ravi Tamada",
                "email": "ravi@gmail.com",
                "address": "xx-xx-xxxx,x - street, x - country",
                "gender" : "male",
                "phone": {
                    "mobile": "+91 0000000000",
          "mobile": "+91 0000000000",
          "home": "00 000000",
          "office": "00 000000"
        }
    },
    {
        "id": "c201",
        "name": "Johnny Depp",
        "email": "johnny_depp@gmail.com",
        "address": "xx-xx-xxxx,x - street, x - country",
        "gender" : "male",
        "phone": {
          "mobile": "+91 0000000000",
          "home": "00 000000",
          "office": "00 000000"
        }
    },
    .
    .
    .
    .
 ]
}
```

The difference between [and { - (Square brackets and Curly brackets)

If you observe normally JSON data will have square brackets and curly brackets. The difference between [and { is, the square bracket represents starting of an **JSONArray** node whereas curly bracket represents **JSONObject**. While accessing these elements we need to call different methods to access these nodes.


JSON Structure
AndroidHive
JSONArray
[- represents its a json Array node
JSONObject
{ - represents its a json Object node

```
{
    "contacts" : [
        {
        "name" : "Ravi Tamada",
        "email" : "ravi8x@gmail.com",
        "gender" : "male",
        "address" : "xx xx xxxxxxx",
        "phone": {
            "mobile": "91 xxxxxxxxxx",
            "office" : "00 00000000"
        },
        {
```

Writing JSON Parser Class

In your project create a class file and name it as **JSONParser.java**. The parser class has a method which will make http request to get JSON data and returns a JSONObject.

```
rser.java
e com.androidhive.jsonparsing;

 java.io.BufferedReader;
 java.io.IOException;
 java.io.InputStream;
 java.io.InputStreamReader;
 java.io.UnsupportedEncodingException;

 org.apache.http.HttpEntity;
 org.apache.http.HttpResponse;
 org.apache.http.client.ClientProtocolException;
 org.apache.http.client.methods.HttpPost;
 org.apache.http.impl.client.DefaultHttpClient;
 org.json.JSONException;
 org.json.JSONObject;

 android.util.Log;

 class JSONParser {

atic InputStream is = null;
atic JSONObject jObj = null;
atic String json = "";

 constructor
blic JSONParser() {



blic JSONObject getJSONFromUrl(String url) {

  // Making HTTP request
  try {
      // defaultHttpClient
      DefaultHttpClient httpClient = new DefaultHttpClient();
      HttpPost httpPost = new HttpPost(url);

      HttpResponse httpResponse = httpClient.execute(httpPost);
      HttpEntity httpEntity = httpResponse.getEntity();
      is = httpEntity.getContent();

  } catch (UnsupportedEncodingException e) {
      e.printStackTrace();
  } catch (ClientProtocolException e) {
      e.printStackTrace();
  } catch (IOException e) {
      e.printStackTrace();
  }

  try {
      BufferedReader reader = new BufferedReader(new InputStreamReader(
              is, "iso-8859-1"), 8);
      StringBuilder sb = new StringBuilder();
      String line = null;
  while ((line = reader.readLine()) != null) {
    sb.append(line + "\n");
  }
  is.close();
  json = sb.toString();
} catch (Exception e) {
  Log.e("Buffer Error", "Error converting result " + e.toString());
}

// try parse the string to a JSON object
try {
  jObj = new JSONObject(json);
} catch (JSONException e) {
  Log.e("JSON Parser", "Error parsing data " + e.toString());
}

// return JSON String
return jObj;
```

Parsing JSON Data

Once you created parser class next thing is to know how to use that class. Below i am explaining how to parse the json (taken in this example) using the parser class.

1. In the contacts json we have items like name, email, address, gender and phone numbers. So first thing is to store all these node names in variables. Open your main activity class and declare store all node names in static variables.

```
// url to make request
private static String url = "http://api.androidhive.info/contacts/";

// JSON Node names
private static final String TAG_CONTACTS = "contacts";
private static final String TAG_ID = "id";
private static final String TAG_NAME = "name";
private static final String TAG_EMAIL = "email";
private static final String TAG_ADDRESS = "address";
private static final String TAG_GENDER = "gender";
private static final String TAG_PHONE = "phone";
private static final String TAG_PHONE_MOBILE = "mobile";
private static final String TAG_PHONE_HOME = "home";
private static final String TAG_PHONE_OFFICE = "office";

// contacts JSONArray
JSONArray contacts = null;
```

2. Next step to is to use parser class to get JSONObject and looping through each json item. Below i am creating an instance of JSONParser class and using for loop i am looping through each json item and finally storing each json data in variable.

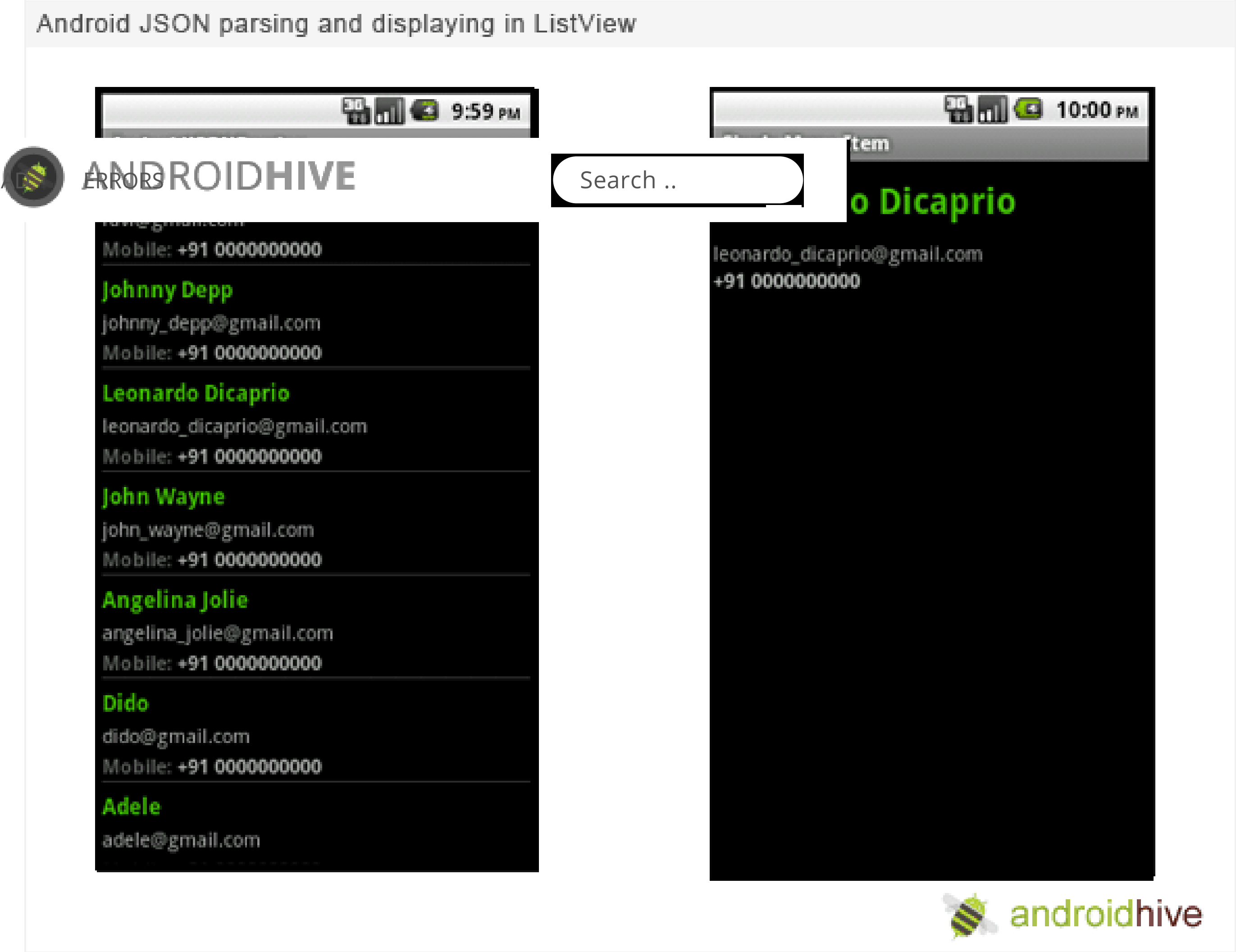
```
// Creating JSON Parser instance
```

```
JSONParser jParser = new JSONParser();

// getting JSON string from URL
JSONObject json = jParser.getJSONFromUrl(url);

try {
    // Getting Array of Contacts
    contacts = json.getJSONArray(TAG_CONTACTS);

    // looping through All Contacts
    for(int i = 0; i < contacts.length(); i++){
        JSONObject c = contacts.getJSONObject(i);

        // Storing each json item in variable
        String id = c.getString(TAG_ID);
        String name = c.getString(TAG_NAME);
        String email = c.getString(TAG_EMAIL);
        String address = c.getString(TAG_ADDRESS);
        String gender = c.getString(TAG_GENDER);

        // Phone number is agin JSON Object
        JSONObject phone = c.getJSONObject(TAG_PHONE);
        String mobile = phone.getString(TAG_PHONE_MOBILE);
        String home = phone.getString(TAG_PHONE_HOME);
        String office = phone.getString(TAG_PHONE_OFFICE);

    }
} catch (JSONException e) {
    e.printStackTrace();
}
```

Parsing JSON data and updating into ListView

In my previous tutorial Android ListView Tutorial i explained how to create listview and updating with list data. Below i am implementing same listview but the list data i am updating is from parsed JSON.

```
AndroidJSONParsingActivity.java
package com.androidhive.jsonparsing;

import java.util.ArrayList;
import java.util.HashMap;

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

import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

public class AndroidJSONParsingActivity extends ListActivity {
    // url to make request
    private static String url = "http://api.androidhive.info/contacts/";

    // JSON Node names
    private static final String TAG_CONTACTS = "contacts";
    private static final String TAG_ID = "id";
    private static final String TAG_NAME = "name";
    private static final String TAG_EMAIL = "email";
    private static final String TAG_ADDRESS = "address";
    private static final String TAG_GENDER = "gender";
    private static final String TAG_PHONE = "phone";
    private static final String TAG_PHONE_MOBILE = "mobile";
    private static final String TAG_PHONE_HOME = "home";
    private static final String TAG_PHONE_OFFICE = "office";

    // contacts JSONArray
    JSONArray contacts = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // Hashmap for ListView
        ArrayList<HashMap<String, String>> contactList = new ArrayList<H

        // Creating JSON Parser instance
        JSONParser jParser = new JSONParser();

        // getting JSON string from URL
        JSONObject json = jParser.getJSONFromUrl(url);

        try {
            // Getting Array of Contacts
            contacts = json.getJSONArray(TAG_CONTACTS);

            // looping through All Contacts
            for(int i = 0; i < contacts.length(); i++){
                JSONObject c = contacts.getJSONObject(i);

                // Storing each json item in variable
                String id = c.getString(TAG_ID);
                String name = c.getString(TAG_NAME);
                String email = c.getString(TAG_EMAIL);
                String address = c.getString(TAG_ADDRESS);
                String gender = c.getString(TAG_GENDER);

                // Phone number is agin JSON Object
                JSONObject phone = c.getJSONObject(TAG_PHONE);
                String mobile = phone.getString(TAG_PHONE_MOBILE);
                String home = phone.getString(TAG_PHONE_HOME);
                String office = phone.getString(TAG_PHONE_OFFICE);

                // creating new HashMap
                HashMap<String, String> map = new HashMap<String, String

                // adding each child node to HashMap key => value
                map.put(TAG_ID, id);
                map.put(TAG_NAME, name);
                map.put(TAG_EMAIL, email);
                map.put(TAG_PHONE_MOBILE, mobile);
                // adding HashList to ArrayList
                contactList.add(map);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }


        /**
         * Updating parsed JSON data into ListView
         * */
        ListAdapter adapter = new SimpleAdapter(this, contactList,
                R.layout.list_item,
                new String[] { TAG_NAME, TAG_EMAIL, TAG_PHONE_MOBILE },
                        R.id.name, R.id.email, R.id.mobile });

        setListAdapter(adapter);

        // selecting single ListView item
        ListView lv = getListView();

        // Launching new screen on Selecting Single ListItem
        lv.setOnItemClickListener(new OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view,
                    int position, long id) {
                // getting values from selected ListItem
                String name = ((TextView) view.findViewById(R.id.name)).
                String cost = ((TextView) view.findViewById(R.id.email))
                String description = ((TextView) view.findViewById(R.id.

                // Starting new intent
                Intent in = new Intent(getApplicationContext(), SingleMe
                in.putExtra(TAG_NAME, name);
                in.putExtra(TAG_EMAIL, cost);
                in.putExtra(TAG_PHONE_MOBILE, description);
                startActivity(in);
            }
        });
    }

}
```



Android JSON parsing and displaying in ListView

```
9:59 PM
ravi@gmail.com
Mobile: +91 0000000000
Johnny Depp
johnny_depp@gmail.com
Mobile: +91 0000000000
Leonardo Dicaprio
leonardo_dicaprio@gmail.com
Mobile: +91 0000000000
John Wayne
john_wayne@gmail.com
Mobile: +91 0000000000
Angelina Jolie
angelina_jolie@gmail.com
Mobile: +91 0000000000
Dido
dido@gmail.com
Mobile: +91 0000000000
Adele
adele@gmail.com
```

```
10:00 PM
tem
o Dicaprio
leonardo_dicaprio@gmail.com
+91 0000000000
```

Share this article on

338 Like

42 Tweet

52

Report a Bug in this article

(If you find any error either in code or content please help me in improvising the content)

Email address / Full name