Android - JSON Parser

Previous Page Next Page

JSON stands for JavaScript Object Notation.It is an independent data exchange format and is the best alternative for XML. This chapter explains how to parse the JSON file and extract necessary information from it.

Android provides four different classes to manipulate JSON data. These classes are **JSONArray,JSONObject,JSONStringer and JSONTokenizer.**

The first step is to identify the fields in the JSON data in which you are interested in. For example. In the JSON given below we interested in getting temperature only.

```
{
   "sys":
   {
      "country":"GB",
      "sunrise":1381107633,
      "sunset":1381149604
   },
   "weather":[
      {
         "id":711,
         "main":"Smoke",
         "description":"smoke",
         "icon":"50n"
      }
   ],

  "main":
   {
      "temp":304.15,
      "pressure":1009,
   }
}
```

JSON - Elements

An JSON file consist of many components. Here is the table defining the components of an JSON file and their description –

Sr.No	Component & description
1	**Array([)** In a JSON file , square bracket ([) represents a JSON array

2	**Objects({}** In a JSON file, curly bracket ({} represents a JSON object
3	**Key** A JSON object contains a key that is just a string. Pairs of key/value make up a JSON object
4	**Value** Each key has a value that could be string , integer or double e.t.c

JSON - Parsing

For parsing a JSON object, we will create an object of class JSONObject and specify a string containing JSON data to it. Its syntax is −

```
String in;
JSONObject reader = new JSONObject(in);
```

The last step is to parse the JSON. A JSON file consist of different object with different key/value pair e.t.c. So JSONObject has a separate function for parsing each of the component of JSON file. Its syntax is given below −

```
JSONObject sys  = reader.getJSONObject("sys");
country = sys.getString("country");

JSONObject main  = reader.getJSONObject("main");
temperature = main.getString("temp");
```

The method **getJSONObject** returns the JSON object. The method **getString** returns the string value of the specified key.

Apart from the these methods , there are other methods provided by this class for better parsing JSON files. These methods are listed below −

Sr.No	Method & description
1	**get(String name)** This method just Returns the value but in the form of Object type
2	**getBoolean(String name)** This method returns the boolean value specified by the key
3	**getDouble(String name)** This method returns the double value specified by the key

4	**getInt(String name)** This method returns the integer value specified by the key
5	**getLong(String name)** This method returns the long value specified by the key
6	**length()** This method returns the number of name/value mappings in this object..
7	**names()** This method returns an array containing the string names in this object.

Example

To experiment with this example , you can run this on an actual device or in an emulator.

Steps	Description
1	You will use Android studio to create an Android application.
2	Modify src/MainActivity.java file to add necessary code.
3	Modify the res/layout/activity_main to add respective XML components
4	Modify the res/values/string.xml to add necessary string components
5	Run the application and choose a running android device and install the application on it and verify the results

Following is the content of the modified main activity file **src/MainActivity.java**.

```
package com.example.tutorialspoint7.myapplication;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

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

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

public class MainActivity extends AppCompatActivity {
```

```
private String TAG = MainActivity.class.getSimpleName();
private ListView lv;

ArrayList<HashMap<String, String>> contactList;

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

   contactList = new ArrayList<>();
   lv = (ListView) findViewById(R.id.list);

   new GetContacts().execute();
}

private class GetContacts extends AsyncTask<Void, Void, Void> {
   @Override
   protected void onPreExecute() {
      super.onPreExecute();
      Toast.makeText(MainActivity.this,"Json Data is
         downloading",Toast.LENGTH_LONG).show();

   }

   @Override
   protected Void doInBackground(Void... arg0) {
      HttpHandler sh = new HttpHandler();
      // Making a request to url and getting response
      String url = "http://api.androidhive.info/contacts/";
      String jsonStr = sh.makeServiceCall(url);

      Log.e(TAG, "Response from url: " + jsonStr);
      if (jsonStr != null) {
         try {
            JSONObject jsonObj = new JSONObject(jsonStr);

            // Getting JSON Array node
            JSONArray contacts = jsonObj.getJSONArray("contacts");

            // looping through All Contacts
            for (int i = 0; i < contacts.length(); i++) {
               JSONObject c = contacts.getJSONObject(i);
               String id = c.getString("id");
               String name = c.getString("name");
               String email = c.getString("email");
               String address = c.getString("address");
               String gender = c.getString("gender");

               // Phone node is JSON Object
               JSONObject phone = c.getJSONObject("phone");
               String mobile = phone.getString("mobile");
               String home = phone.getString("home");
               String office = phone.getString("office");

               // tmp hash map for single contact
               HashMap<String, String> contact = new HashMap<>();

               // adding each child node to HashMap key => value
               contact.put("id", id);
               contact.put("name", name);
               contact.put("email", email);
               contact.put("mobile", mobile);
```

```
                // adding contact to contact list
                contactList.add(contact);
            }
         } catch (final JSONException e) {
            Log.e(TAG, "Json parsing error: " + e.getMessage());
            runOnUiThread(new Runnable() {
               @Override
               public void run() {
                  Toast.makeText(getApplicationContext(),
                  "Json parsing error: " + e.getMessage(),
                     Toast.LENGTH_LONG).show();
               }
            });

         }

      } else {
         Log.e(TAG, "Couldn't get json from server.");
         runOnUiThread(new Runnable() {
            @Override
            public void run() {
               Toast.makeText(getApplicationContext(),
                  "Couldn't get json from server. Check LogCat for possible errors!
                  Toast.LENGTH_LONG).show();
            }
         });
      }

      return null;
   }

   @Override
   protected void onPostExecute(Void result) {
      super.onPostExecute(result);
      ListAdapter adapter = new SimpleAdapter(MainActivity.this, contactList,
         R.layout.list_item, new String[]{ "email","mobile"},
            new int[]{R.id.email, R.id.mobile});
      lv.setAdapter(adapter);
   }
}
}
```

Following is the modified content of the xml **HttpHandler.java**.

```
package com.example.tutorialspoint7.myapplication;

import android.util.Log;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.ProtocolException;
import java.net.URL;

public class HttpHandler {

   private static final String TAG = HttpHandler.class.getSimpleName();

   public HttpHandler() {
```

```
    }

    public String makeServiceCall(String reqUrl) {
        String response = null;
        try {
            URL url = new URL(reqUrl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            // read the response
            InputStream in = new BufferedInputStream(conn.getInputStream());
            response = convertStreamToString(in);
        } catch (MalformedURLException e) {
            Log.e(TAG, "MalformedURLException: " + e.getMessage());
        } catch (ProtocolException e) {
            Log.e(TAG, "ProtocolException: " + e.getMessage());
        } catch (IOException e) {
            Log.e(TAG, "IOException: " + e.getMessage());
        } catch (Exception e) {
            Log.e(TAG, "Exception: " + e.getMessage());
        }
        return response;
    }

    private String convertStreamToString(InputStream is) {
        BufferedReader reader = new BufferedReader(new InputStreamReader(is));
        StringBuilder sb = new StringBuilder();

        String line;
        try {
            while ((line = reader.readLine()) != null) {
                sb.append(line).append('\n');
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                is.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        return sb.toString();
    }
}
```

Following is the modified content of the xml **res/layout/activity_main.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.tutorialspoint7.myapplication.MainActivity">

    <ListView
        android:id="@+id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

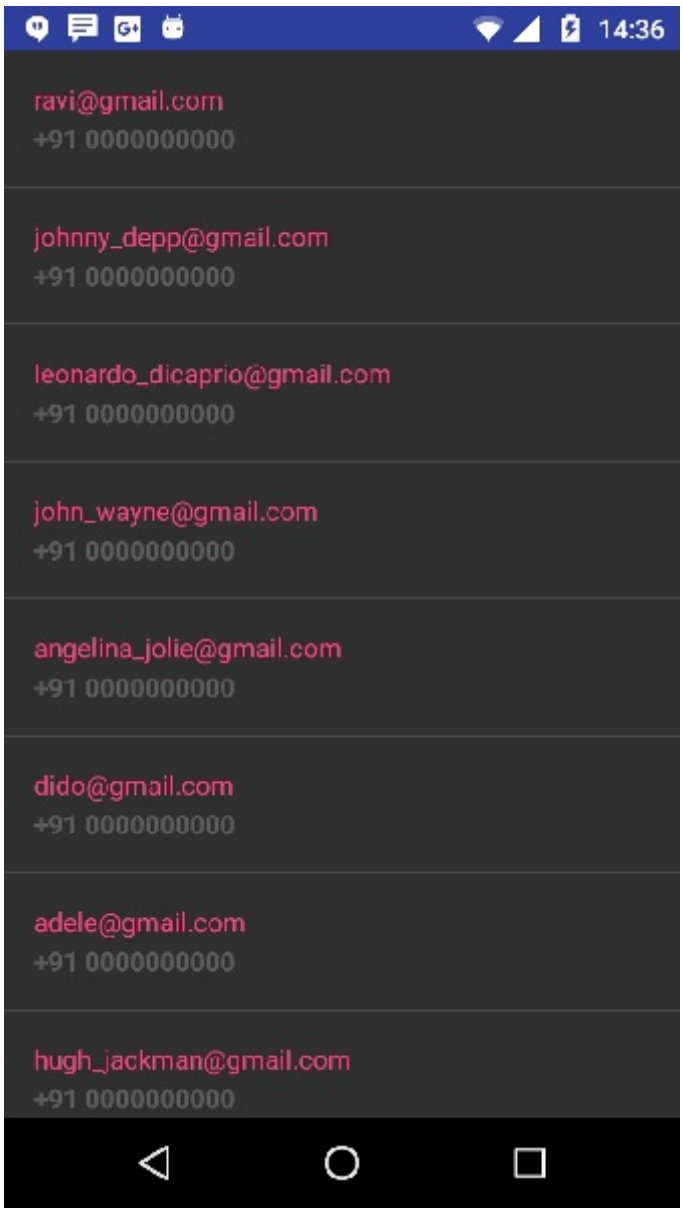
Following is the modified content of the xml **res/layout/list_item.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical"
   android:padding="@dimen/activity_horizontal_margin">
   <TextView
      android:id="@+id/email"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:paddingBottom="2dip"
      android:textColor="@color/colorAccent" />

   <TextView
      android:id="@+id/mobile"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textColor="#5d5d5d"
      android:textStyle="bold" />
</LinearLayout>
```

Following is the content of **AndroidManifest.xml** file.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tutorialspoint7.myapplication">

   <uses-permission android:name="android.permission.INTERNET"/>
   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">
         <activity android:name=".MainActivity">
            <intent-filter>
               <action android:name="android.intent.action.MAIN" />
               <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Let's try to run our application we just modified. I assume you had created your **AVD** while doing environment setup. To run the app from Android studio, open one of your project's activity files and click Run icon from the toolbar. Android studio installs the app on your AVD and starts it and if everything is fine with your setup and application, it will display following Emulator window −



Above Example showing the data from string json,The data has contained employer details as well as salary information.

Previous Page Next Page

Enter email for newsletter go