Handheld Application Development

Lec 13: Network I: JSON

Ekarat Rattagan, PhD

JSON

JavaScript Object Notation (JSON) – language independent specification of interchanging data in human-readable format.

•In contrast with XML-RPC and SOAP, it allows for bidirectional communication between client and server (like peers) and for multiple calls.

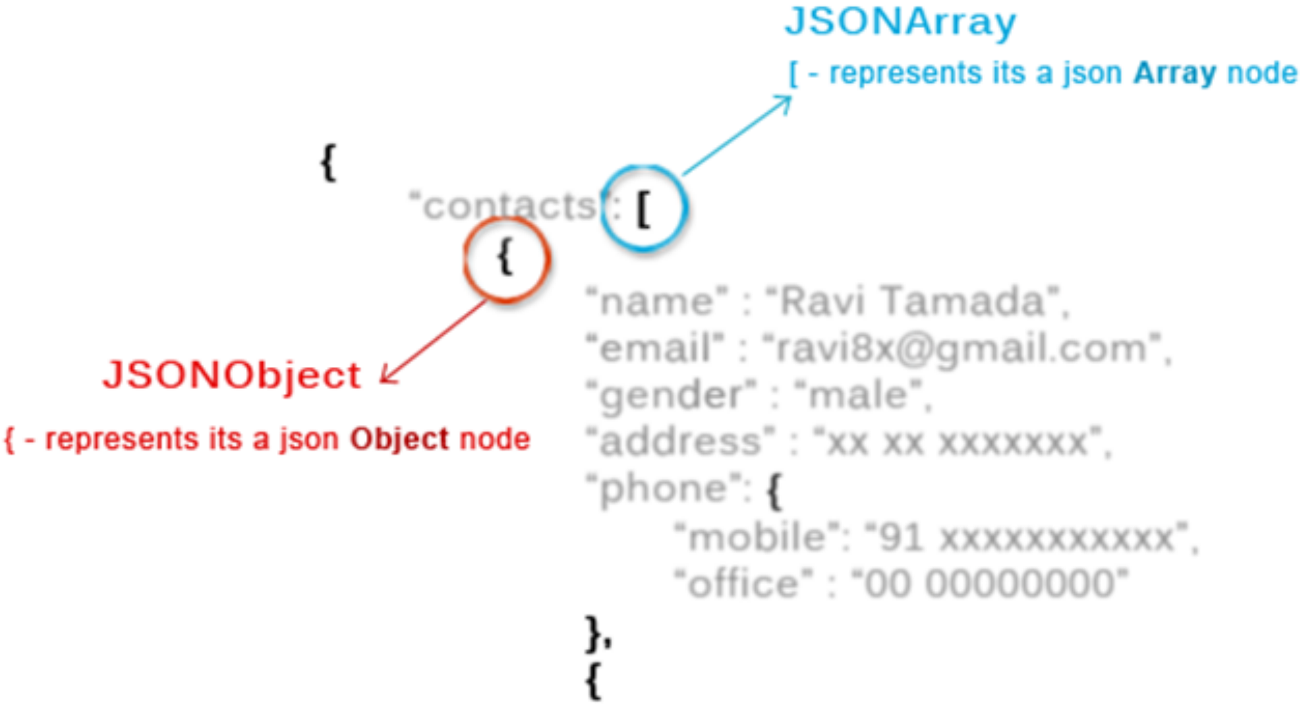
•Data introduced in key:value pair format.

•More economical than XML in terms of data size, but has less readability.

•Android has JSONArray, JSONObject, JSONStringer and JSONTokener objects for parsing this format.

JSON symbols

“[“ means an array, “{“ means an object.



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 xxxxxxxxxxx”,
                “office” : “00 00000000”
            }
        },
        {
```

The example of JSON

A list of contacts where each node contains contact information like

id,

name,

email,

address,

gender,

phone numbers.

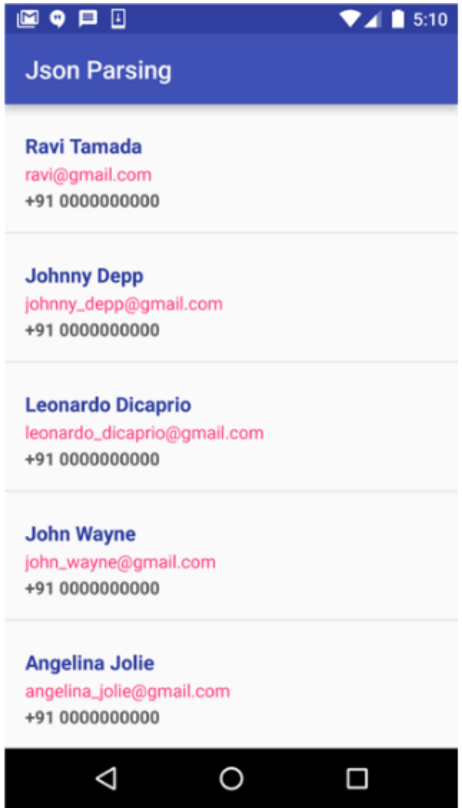
```
{
    "contacts": [
        {
                "id": "c200",
                "name": "Ravi Tamada",
                "email": "ravi@gmail.com",
                "address": "xx-xx-xxxx,x - street, x - country",
                "gender" : "male",
                "phone": {
                    "mobile": "+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"
                }
        }
     ]
}
```

https://api.androidhive.info/contacts/

The sample JSON

Example of JSON

- http://api.androidhive.info/contacts/



Ravi Tamada
ravi@gmail.com
+91 0000000000

Johnny Depp
johnny_depp@gmail.com
+91 0000000000

Leonardo Dicaprio
leonardo_dicaprio@gmail.com
+91 0000000000

John Wayne
john_wayne@gmail.com
+91 0000000000

Angelina Jolie
angelina_jolie@gmail.com
+91 0000000000

1. Create MainActivity

```
package Your package name

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();
  }
```

Create GetContacts AsyncTask (1/3)

```
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);

    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");
```

Create GetContacts AsyncTask (2/3)

```
    // 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);
  } // end for
} // end try
```

Create GetContacts AsyncTask (3/3)

```
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);
    }
  }
}
```

Create HttpHandler class (1/2)

```
package [Your package name]

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;
    }
```

Create HttpHandler class (2/2)

```
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();
  }
}
```

Create ListView

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>
```

Create list_item view

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>
```