|  |  |  |  |
| --- | --- | --- | --- |
| Exercise: Android Networking | | | |
| Document Number: |  | Issue Number:  Date: |  |

TABLE OF CONTENTS

[**1.** **Web Service Overview** 2](#_Toc366847772)

[**2.** **Application Progress Flow** 3](#_Toc366847773)

[**3.** **Implementation** 6](#_Toc366847774)

[**4.** **Advanced** 20](#_Toc366847775)

# **Web Service Overview**

***Here are the code and snapshots for creating a ListView with dynamic data retrieved from*** <http://api.androidhive.info/music/music.xml>***:***

Sample XML data:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<music>

<song>

<id>1</id>

<title>Someone Like You</title>

<artist>Adele</artist>

<duration>4:47</duration>

<plays>1662</plays>

<thumb\_url>http://api.androidhive.info/music/images/adele.png</thumb\_url>

</song>

<song>

<id>2</id>

<title>Space Bound</title>

<artist>Eminem</artist>

<duration>4:38</duration>

<plays>1900</plays>

<thumb\_url>http://api.androidhive.info/music/images/eminem.png</thumb\_url>

</song>

.

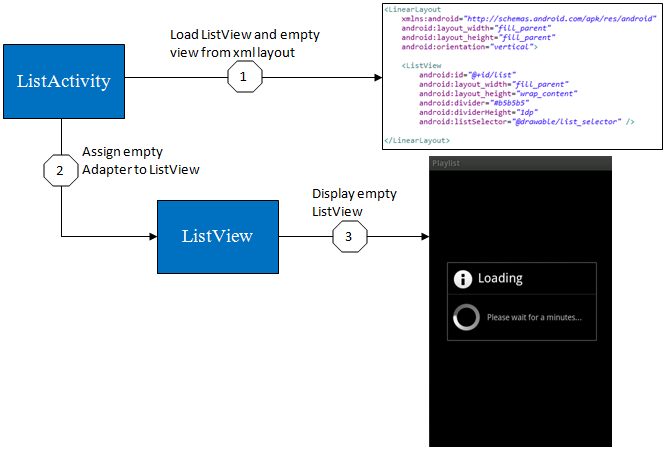
.

.

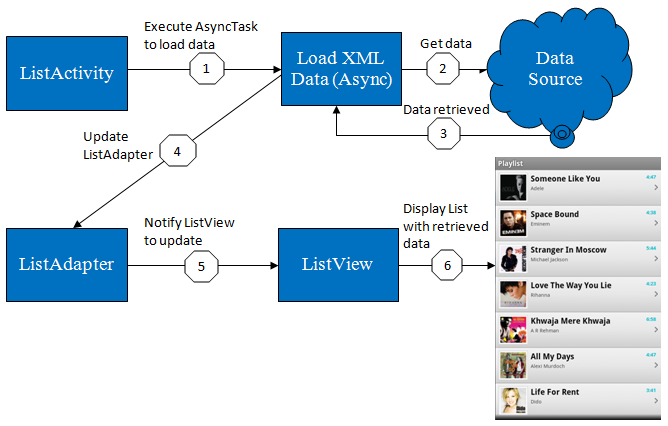
.

</music>

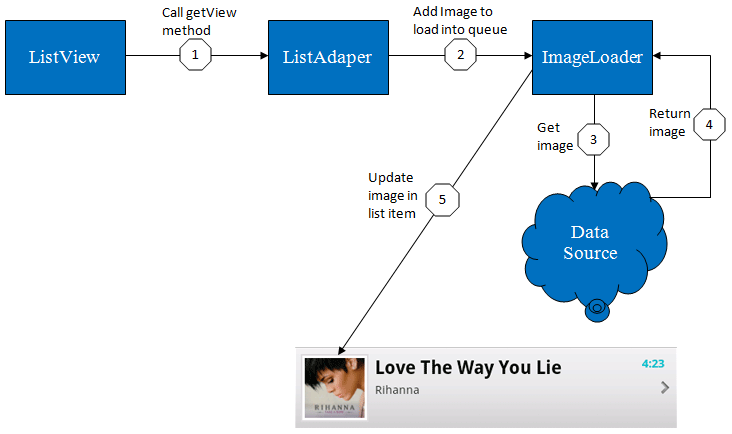
# **Application Progress Flow**



**Initializing an empty Listview**

****

**Loading data with AsyncTask**

****

**Loading image for each list item**

# **Implementation**

Follow the steps described in this Android Tutorial:

**Step 1:** Create new project in your Eclipse IDE and fill all the details. **File ⇒ New Project.**

**Step 2:** Right Click and **drawable-hdpi ⇒ New ⇒ Android XML File:** gradient\_bg.xml, list\_selector.xml, gradient\_bg\_hover.xml, image\_bg.xml.

**gradient\_bg.xml** – Default Background Gradient Style

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<shape xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:shape=*"rectangle"*>

<!-- Gradient Bg for listrow -->

<gradient

android:startColor=*"#f1f1f2"*

android:centerColor=*"#e7e7e8"*

android:endColor=*"#cfcfcf"*

android:angle=*"270"* />

</shape>

**gradient\_bg\_hover.xml** – Gradient Style for on hover state

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<shape xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:shape=*"rectangle"*>

<!-- Gradient BgColor for listrow Selected -->

<gradient

android:startColor=*"#18d7e5"*

android:centerColor=*"#16cedb"*

android:endColor=*"#09adb9"*

android:angle=*"270"* />

</shape>

**list\_selector.xml** – Actual liststyle which combines the above two styles

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<selector xmlns:android=*"http://schemas.android.com/apk/res/android"*>

<!-- Selector style for listrow -->

<item

android:state\_selected=*"false"*

android:state\_pressed=*"false"*

android:drawable=*"@drawable/gradient\_bg"* />

<item android:state\_pressed=*"true"*

android:drawable=*"@drawable/gradient\_bg\_hover"* />

<item android:state\_selected=*"true"*

android:state\_pressed=*"false"*

android:drawable=*"@drawable/gradient\_bg\_hover"* />

</selector>

**image\_bg.xml** – is for white border around the image in listview

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<layer-list xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item>

<shape

android:shape=*"rectangle"*>

<stroke android:width=*"1dp"* android:color=*"#dbdbdc"* />

<solid android:color=*"#FFFFFF"* />

</shape>

</item>

</layer-list>

Now open your **main.xml** file and define a listview in it.

<?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=*"fill\_parent"*

android:orientation=*"vertical"*>

<ListView

android:id=*"@+id/list"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

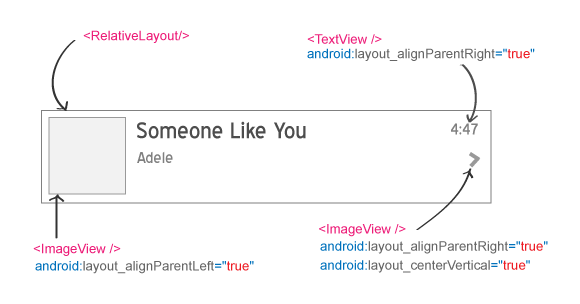
android:divider=*"#b5b5b5"*

android:dividerHeight=*"1dp"*

android:listSelector=*"@drawable/list\_selector"* />

</LinearLayout>

**Step 3:** Next step is to design single listrow. Create a new XML file under layout folder and name it as list\_row.xml. **Right Click ⇒ New ⇒ Android XML File.**

****

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:background=*"@drawable/list\_selector"*

android:orientation=*"horizontal"*

android:padding=*"5dip"* >

<!-- ListRow Left sied Thumbnail image -->

<LinearLayout

android:id=*"@+id/thumbnail"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_marginRight=*"5dip"*

android:background=*"@drawable/image\_bg"*

android:padding=*"3dip"* >

<ImageView

android:id=*"@+id/list\_image"*

android:layout\_width=*"50dip"*

android:layout\_height=*"50dip"*

android:src=*"@drawable/rihanna"* />

</LinearLayout>

<!-- Title Of Song -->

<TextView

android:id=*"@+id/title"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignTop=*"@+id/thumbnail"*

android:layout\_toRightOf=*"@+id/thumbnail"*

android:text=*"Rihanna Love the way lie"*

android:textColor=*"#040404"*

android:textSize=*"15dip"*

android:textStyle=*"bold"*

android:typeface=*"sans"* />

<!-- Artist Name -->

<TextView

android:id=*"@+id/artist"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@id/title"*

android:layout\_marginTop=*"1dip"*

android:layout\_toRightOf=*"@+id/thumbnail"*

android:text=*"Just gona stand there and ..."*

android:textColor=*"#343434"*

android:textSize=*"10dip"* />

<!-- Rightend Duration -->

<TextView

android:id=*"@+id/duration"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentRight=*"true"*

android:layout\_alignTop=*"@id/title"*

android:layout\_marginRight=*"5dip"*

android:gravity=*"right"*

android:text=*"5:45"*

android:textColor=*"#10bcc9"*

android:textSize=*"10dip"*

android:textStyle=*"bold"* />

<!-- Rightend Arrow -->

<ImageView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

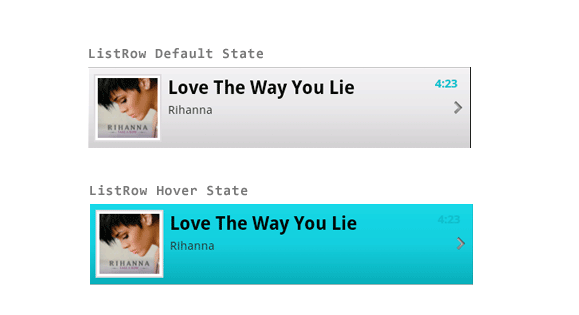
android:layout\_alignParentRight=*"true"*

android:layout\_centerVertical=*"true"*

android:src=*"@drawable/arrow"* />

</RelativeLayout>

The above XML will provide you output like below:



**Step 4:** Next step is to parse the xml and update the data into listview. Create a new java class file in your src folder. **Right Click on src ⇒ New ⇒ Class** and name it as LazyAdapter.java

**package** com.csc.example.networking;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** com.csc.example.networking.R;

**import** android.app.Activity;

**import** android.content.Context;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.BaseAdapter;

**import** android.widget.ImageView;

**import** android.widget.TextView;

**public** **class** LazyAdapter **extends** BaseAdapter {

**private** Activity activity;

**private** ArrayList<HashMap<String, String>> data;

**private** **static** LayoutInflater *inflater*=**null**;

**public** ImageLoader imageLoader;

**public** LazyAdapter(Activity a, ArrayList<HashMap<String, String>> d) {

activity = a;

data = d;

*inflater* = (LayoutInflater)activity

.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);

imageLoader = **new** ImageLoader(activity.getApplicationContext());

}

**public** **int** getCount() {

**return** data.size();

}

**public** Object getItem(**int** position) {

**return** position;

}

**public** **long** getItemId(**int** position) {

**return** position;

}

**public** View getView(**int** position, View convertView, ViewGroup parent) {

View vi=convertView;

**if**(convertView==**null**)

vi = *inflater*.inflate(R.layout.*list\_row*, **null**);

TextView title = (TextView)vi.findViewById(R.id.*title*);

TextView artist = (TextView)vi.findViewById(R.id.*artist*);

TextView duration = (TextView)vi.findViewById(R.id.*duration*);

ImageView thumb\_image = (ImageView)vi.findViewById(R.id.*list\_image*);

HashMap<String, String> song = **new** HashMap<String, String>();

song = data.get(position);

// Setting all values in listview

title.setText(song.get(MainActivity.*KEY\_TITLE*));

artist.setText(song.get(MainActivity.*KEY\_ARTIST*));

duration.setText(song.get(MainActivity.*KEY\_DURATION*));

imageLoader.DisplayImage(song.get(MainActivity.*KEY\_THUMB\_URL*), thumb\_image);

**return** vi;

}

}

**Step 5:** Now open your **MainActivity class** and type the following code. In the following code i am getting xml from url and parsing it. While parsing i am storing all the xml data into HashMap and finally i am passing HashMap to LazyAdapter class.

**package** com.csc.example.networking;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.UnsupportedEncodingException;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** org.apache.http.HttpEntity;

**import** org.apache.http.HttpResponse;

**import** org.apache.http.client.ClientProtocolException;

**import** org.apache.http.client.HttpClient;

**import** org.apache.http.client.methods.HttpGet;

**import** org.apache.http.impl.client.DefaultHttpClient;

**import** org.xmlpull.v1.XmlPullParserException;

**import** com.csc.example.networking.R;

**import** android.app.Activity;

**import** android.app.ProgressDialog;

**import** android.os.AsyncTask;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.AdapterView;

**import** android.widget.AdapterView.OnItemClickListener;

**import** android.widget.ListView;

**public** **class** MainActivity **extends** Activity {

// All static variables

**public** **static** **final** String *URL* = http://api.androidhive.info/music/music.xml";

// XML node keys

**public** **static** **final** String *KEY\_MUSIC* = "music"; // root node

**public** **static** **final** String *KEY\_SONG* = "song"; // parent node

**public** **static** **final** String *KEY\_ID* = "id";

**public** **static** **final** String *KEY\_TITLE* = "title";

**public** **static** **final** String *KEY\_ARTIST* = "artist";

**public** **static** **final** String *KEY\_DURATION* = "duration";

**public** **static** **final** String *KEY\_THUMB\_URL* = "thumb\_url";

ProgressDialog progressDialog;

ListView list;

LazyAdapter adapter;

ArrayList<HashMap<String, String>> songsList

= **new** ArrayList<HashMap<String, String>>();

@Override

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

list = (ListView) findViewById(R.id.*list*);

// Assign empty adapter

adapter = **new** LazyAdapter(**this**, songsList);

list.setAdapter(adapter);

// Click event for single list row

list.setOnItemClickListener(**new** OnItemClickListener() {

@Override

**public** **void** onItemClick(AdapterView<?> parent, View view

, **int** position, **long** id) {

}

});

downloadXmlData();

}

**private** **void** downloadXmlData() {

progressDialog = **new** ProgressDialog(MainActivity.**this**);

progressDialog.setTitle("Loading");

progressDialog.setMessage("Please wait for a minutes...");

progressDialog.show();

DownloadTask downloadTask = **new** DownloadTask();

downloadTask.execute(*URL*);

}

**private** **class** DownloadTask **extends** AsyncTask<String, Void,

List<HashMap<String, String>>> {

XMLParser parser = **new** XMLParser();

@Override

**protected** List<HashMap<String, String>>

doInBackground(String... params) {

// getting XML from URL

**try** {

HttpClient httpClient = **new** DefaultHttpClient();

HttpGet httpGet = **new** HttpGet(params[0]);

HttpResponse httpResponse = httpClient.execute(httpGet);

HttpEntity httpEntity = httpResponse.getEntity();

InputStream is = httpEntity.getContent();

**return** parser.parse(is);

}

**catch** (UnsupportedEncodingException e) {

}

**catch** (ClientProtocolException e) {

}

**catch** (IOException e) {

}

**catch** (XmlPullParserException e) { }

**return** **null**;

}

@Override

**protected** **void** onPostExecute(List<HashMap<String, String>> result) {

**try** {

songsList.clear();

songsList.addAll(result);

adapter.notifyDataSetChanged();

}

**catch** (Exception ex) {

}

**finally** {

progressDialog.hide();

}

}

}

@Override

**public** **void** onDestroy(){

**super**.onDestroy();

**if** (progressDialog!=**null**) {

progressDialog.cancel();

}

}

}

**Step 6**: Create XMLParser.class

**XMLParser.java**

**package** com.csc.example.networking;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** org.xmlpull.v1.XmlPullParser;

**import** org.xmlpull.v1.XmlPullParserException;

**import** android.util.Xml;

**public** **class** XMLParser {

**public** List<HashMap<String, String>> parse(InputStream in)

**throws** XmlPullParserException, IOException {

**try** {

XmlPullParser parser = Xml.*newPullParser*();

parser.setFeature(XmlPullParser

.*FEATURE\_PROCESS\_NAMESPACES*, **false**);

parser.setInput(in, **null**);

parser.nextTag();

**return** readPlaylist(parser);

}

**finally** {

in.close();

}

}

**private** List<HashMap<String, String>> readPlaylist(XmlPullParser parser)

**throws** XmlPullParserException, IOException {

List<HashMap<String, String>> entries =

**new** ArrayList<HashMap<String, String>>();

parser.require(XmlPullParser.*START\_TAG*, **null**, "music");

**while** (parser.next() != XmlPullParser.*END\_TAG*) {

**if** (parser.getEventType() != XmlPullParser.*START\_TAG*) {

**continue**;

}

String name = parser.getName();

// Starts by looking for the entry tag

**if** (name.equals(MainActivity.*KEY\_SONG*))

entries.add(readSong(parser));

**else**

skip(parser);

}

**return** entries;

}

// Parses the contents of an playlist.

**private** HashMap<String, String> readSong(XmlPullParser parser)

**throws** XmlPullParserException, IOException {

HashMap<String, String> song = **new** HashMap<String, String>();

parser.require(XmlPullParser.*START\_TAG*, **null**, MainActivity.*KEY\_SONG*);

String id = **null**;

String title = **null**;

String artist = **null**;

String duration = **null**;

String thumb\_url = **null**;

**while** (parser.next() != XmlPullParser.*END\_TAG*) {

**if** (parser.getEventType() != XmlPullParser.*START\_TAG*) {

**continue**;

}

String name = parser.getName();

**if** (name.equals(MainActivity.*KEY\_ID*))

id = readTag(parser, MainActivity.*KEY\_ID*);

**else** **if** (name.equals(MainActivity.*KEY\_TITLE*))

title = readTag(parser, MainActivity.*KEY\_TITLE*);

**else** **if** (name.equals(MainActivity.*KEY\_ARTIST*))

artist = readTag(parser, MainActivity.*KEY\_ARTIST*);

**else** **if** (name.equals(MainActivity.*KEY\_DURATION*))

duration = readTag(parser, MainActivity.*KEY\_DURATION*);

**else** **if** (name.equals(MainActivity.*KEY\_THUMB\_URL*))

thumb\_url = readTag(parser, MainActivity.*KEY\_THUMB\_URL*);

**else**

skip(parser);

}

song.put(MainActivity.*KEY\_ID*, id);

song.put(MainActivity.*KEY\_TITLE*, title);

song.put(MainActivity.*KEY\_ARTIST*, artist);

song.put(MainActivity.*KEY\_DURATION*, duration);

song.put(MainActivity.*KEY\_THUMB\_URL*, thumb\_url);

**return** song;

}

// Processes tags in the playlist.

**private** String readTag(XmlPullParser parser, String tag)

**throws** IOException, XmlPullParserException {

parser.require(XmlPullParser.*START\_TAG*, **null**, tag);

String title = readText(parser);

parser.require(XmlPullParser.*END\_TAG*, **null**, tag);

**return** title;

}

// Extracts text values.

**private** String readText(XmlPullParser parser)

**throws** IOException, XmlPullParserException {

String result = "";

**if** (parser.next() == XmlPullParser.*TEXT*) {

result = parser.getText();

parser.nextTag();

}

**return** result;

}

// Skips tags the parser isn't interested in. Uses depth to

// handle nested tags. i.e.,

// if the next tag after a START\_TAG isn't a matching END\_TAG,

// it keeps going until it

// finds the matching END\_TAG (as indicated by the value of "depth" being 0).

**private** **void** skip(XmlPullParser parser)

**throws** XmlPullParserException, IOException {

**if** (parser.getEventType() != XmlPullParser.*START\_TAG*) {

**throw** **new** IllegalStateException(); }

**int** depth = 1;

**while** (depth != 0) {

**switch** (parser.next()) {

**case** XmlPullParser.*END\_TAG*:

depth--;

**break**;

**case** XmlPullParser.*START\_TAG*:

depth++;

**break**;

}

}

}

}

**Step 7: Open your** AndroidManifest.xml **file add two permissons** INTERNET **and** WRITE\_EXTERNAL\_STORAGE**.**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*

package=*"com.csc.example.networking"* android:versionCode=*"1"*

android:versionName=*"1.0"*>

<uses-sdk android:minSdkVersion=*"8"* android:targetSdkVersion=*"10"*/>

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

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"* />

<application

android:icon=*"@drawable/icon"*

android:label=*"@string/app\_name"*>

<activity

android:name=*"com.csc.example.networking.MainActivity"*

android:label=*"@string/app\_name"*>

<intent-filter>

<action

android:name=*"android.intent.action.MAIN"* />

<category

android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

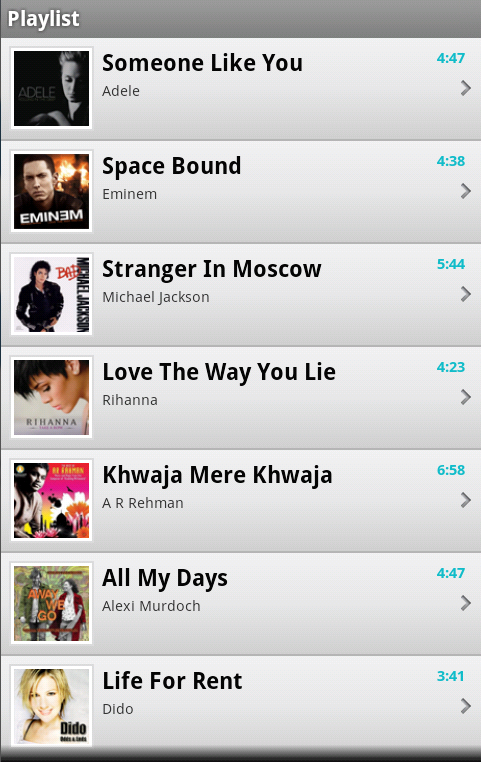
</activity>

</application>

</manifest>

**Step 8:** Import the ImageDownloader library to your project

**Step 9:** After adding all the classes if you run the project it will show output like below:



# **Advanced**

Open ImageDownloader project and find out how the image downloading progress is working via these files:

* **ImageLoader.java**
* **MemoryCache.java**
* **FileCache.java**
* **Utils.java**