Android Upload Image Using PHP MySQL and Display Images in ListView

September 20, 2015 by Belal Khan — 171 Comments

Hello friends welcome to our new Android Upload Image Using PHP MySQL Tutorial. I already published tutorials about how to upload image to server from android. But in last tutorial I stored android images to MySQL database. Now in this tutorial we won't store image to MySQL database instead we will save our image inside a directory. So in this tutorial we will store image to a directory and the path of the image to our mysql database. And we will also fetch the stored image to our Android App. So lets begin.

Android Upload Image Using PHP MySQL Video

You can check out this video before moving ahead. This will show you that what you will be creating with this tutorial.

database. Create the following database

android upload image database

- Now, in your server create a new directory. I created PhotoUpload
- Inside the directory you need a script to connect to your database. So create a script name dbConnect.php and write the following code

Database Connection PHP

```php
<?php
 define('HOST','mysql.hostinger.in');
 define('USER','u502452270_andro');
 define('PASS','belal_123');
 define('DB','u502452270_andro');

 $con = mysqli_connect(HOST,USER,PASS,DB) or die('Unable to Connect');
```

- Now you need one more script that will handle the photo upload. So I created a script named upload.php with the following code

Android Upload Image Script PHP

```php
<?php

 if($_SERVER['REQUEST_METHOD']=='POST'){

 $image = $_POST['image'];

 require_once('dbConnect.php');

 $sql ="SELECT id FROM photos ORDER BY id ASC";

 $res = mysqli_query($con,$sql);

 $id = 0;

 while($row = mysqli_fetch_array($res)){
```

```
$path = "uploads/$id.png";

$actualpath = "http://simplifiedcoding.16mb.com/PhotoUpload/$path";

$sql = "INSERT INTO photos (image) VALUES ('$actualpath')";

if(mys
file_put_contents($path,base64_decode($image));
echo "Successfully Uploaded";
}

mysqli_close($con);
}else{
echo "Error";
}
```

- Now the above script will handle uploads. It will store the images sent to a directory name uploads. So you also need to create a directory name uploads.
- The above script will store the path to the images inside the MySQL database through which we will fetch all the images.
- So we need one more image which will give the urls of all the images. For getting the urls we will use JSON.
- Create a new script named getAllImages.php and write the following code

Android Get All Images from Server PHP

```
<?php
require_once('dbConnect.php');

$sql = "select image from photos";

$res = mysqli_query($con,$sql);

$result = array();

while($row = mysqli_fetch_array($res)){
array_push($result,array('url'=>$row['image']));
}
```

Android Upload Image to Server Using PHP MySQL

Android TabLayout Example using ViewPager and Fragments

Firebase Cloud Messaging Tutorial for Android

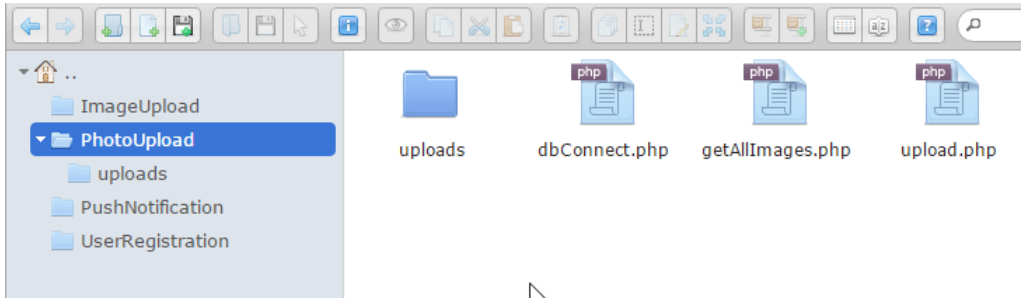
Android Push Noti cation using GCM Tutorial

Android Volley Tutorial – User Registration and Login

Retrieve Data From MySQL Database in Android using Volley

```
16  mysqli_close($con);
```

- Now our server part is over. You can see below the snapshot of my server's directory.



Android Upload Image To Server Using PHP MySQL

- I have the path of my upload.php and getAllImages.php script.
- upload.php -> http://simpli edcoding.16mb.com/PhotoUpload/upload.php
- getAllImages.php -> http://simpli edcoding.16mb.com/PhotoUpload/getAllImages.php

Creating Android Upload Image Using PHP MySQL Project

- Open Android Studio and Create a New Android Project
- This part is same as we did before in how to upload image from android
- You need to create the following layout

Android Upload Image – Main Activity Layout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">


    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/string_choose_file"
        android:id="@+id/buttonChoose" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/imageView" />


    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/string_upload_image"
        android:id="@+id/buttonUpload" />


    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/string_view_image"
        android:id="@+id/buttonViewImage" />


</LinearLayout>
```

```java
package net.simplifiedcoding.imageuploadsample;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;

import javax.net.ssl.HttpsURLConnection;

/**
 * Created by Belal on 8/19/2015.
 */

public class RequestHandler {

    public String sendPostRequest(String requestURL,
                                  HashMap<String, String> postDataParams) {

        URL url;

        StringBuilder sb = new StringBuilder();
        try {
            url = new URL(requestURL);

            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(15000);
            conn.setConnectTimeout(15000);
            conn.setRequestMethod("POST");
            conn.setDoInput(true);
            conn.setDoOutput(true);

```

```
                    new OutputStreamWriter(os, "UTF-8"));
            writer.write(getPostDataString(postDataParams));

            writer.flush();
            writer.close();
            os.close();
            int responseCode = conn.getResponseCode();

            if (responseCode == HttpsURLConnection.HTTP_OK) {
                BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream(
                sb = new StringBuilder();
                String response;
                while ((response = br.readLine()) != null){
                    sb.append(response);
                }
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return sb.toString();
    }

    private String getPostDataString(HashMap<String, String> params) throws UnsupportedEncodingE
        StringBuilder result = new StringBuilder();
        boolean first = true;
        for (Map.Entry<String, String> entry : params.entrySet()) {
            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
        }

        return result.toString();
    }
```

- Now in the MainActivity.java write the following code to make the upload work.

Android Upload Image Using PHP MySQL

```
1  package net.simplifiedcoding.imageuploadsample;
2
3  import android.app.ProgressDialog;
4  import android.content.Context;
5  import android.content.Intent;
6  import android.database.Cursor;
7  import android.graphics.Bitmap;
8  import android.net.Uri;
9  import android.os.AsyncTask;
10 import android.provider.MediaStore;
11 import android.support.v7.app.AppCompatActivity;
12 import android.os.Bundle;
13 import android.util.Base64;
14 import android.util.Log;
15 import android.view.Menu;
16 import android.view.MenuItem;
17 import android.view.View;
18 import android.widget.Button;
19 import android.widget.ImageView;
20 import android.widget.Toast;
21
22 import java.io.ByteArrayOutputStream;
23 import java.io.IOException;
24 import java.util.HashMap;
25
26 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
27
28     public static final String UPLOAD_URL = "http://simplifiedcoding.16mb.com/PhotoUpload/uploa
29     public static final String UPLOAD_KEY = "image";
30
31
32
33     private int PICK_IMAGE_REQUEST = 1;
34
35     private Button buttonChoose;
36     private Button buttonUpload;
```

```
40
41      private Bitmap bitmap;
42
43      private Uri filePath;
44
45      @Override
46      protected void onCreate(Bundle savedInstanceState) {
47          super.onCreate(savedInstanceState);
48          setContentView(R.layout.activity_main);
49
50          buttonChoose = (Button) findViewById(R.id.buttonChoose);
51          buttonUpload = (Button) findViewById(R.id.buttonUpload);
52          buttonView = (Button) findViewById(R.id.buttonViewImage);
53
54          imageView = (ImageView) findViewById(R.id.imageView);
55
56          buttonChoose.setOnClickListener(this);
57          buttonUpload.setOnClickListener(this);
58          buttonView.setOnClickListener(this);
59      }
60
61      private void showFileChooser() {
62          Intent intent = new Intent();
63          intent.setType("image/*");
64          intent.setAction(Intent.ACTION_GET_CONTENT);
65          startActivityForResult(Intent.createChooser(intent, "Select Picture"), PICK_IMAGE_REQUES
66      }
67
68      @Override
69      protected void onActivityResult(int requestCode, int resultCode, Intent data) {
70          super.onActivityResult(requestCode, resultCode, data);
71
72          if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null && data
73
74              filePath = data.getData();
75              try {
76                  bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), filePath);
77                  imageView.setImageBitmap(bitmap);
78              } catch (IOException e) {
```

```
82      }
83
84      public String getStringImage(Bitmap bmp){
85          ByteArrayOutputStream baos = new ByteArrayOutputStream();
86          bmp.compress(Bitmap.CompressFormat.JPEG, 100, baos);
87          byte[] imageBytes = baos.toByteArray();
88          String encodedImage = Base64.encodeToString(imageBytes, Base64.DEFAULT);
89          return encodedImage;
90      }
91
92      private void uploadImage(){
93          class UploadImage extends AsyncTask<Bitmap,Void,String>{
94
95              ProgressDialog loading;
96              RequestHandler rh = new RequestHandler();
97
98              @Override
99              protected void onPreExecute() {
100                 super.onPreExecute();
101                 loading = ProgressDialog.show(MainActivity.this, "Uploading...", null,true,true
102             }
103
104             @Override
105             protected void onPostExecute(String s) {
106                 super.onPostExecute(s);
107                 loading.dismiss();
108                 Toast.makeText(getApplicationContext(),s,Toast.LENGTH_LONG).show();
109             }
110
111             @Override
112             protected String doInBackground(Bitmap... params) {
113                 Bitmap bitmap = params[0];
114                 String uploadImage = getStringImage(bitmap);
115
116                 HashMap<String,String> data = new HashMap<>();
117
118                 data.put(UPLOAD_KEY, uploadImage);
119                 String result = rh.sendPostRequest(UPLOAD_URL,data);
120
```

```
124
125         UploadImage ui = new UploadImage();
126         ui.execute(bitmap);
127     }
128
129     @Override
130     public void onClick(View v) {
131         if (v == buttonChoose) {
132             showFileChooser();
133         }
134
135         if(v == buttonUpload){
136             uploadImage();
137         }
138
139         if(v == buttonView){
140             viewImage();
141         }
142     }
143
144     private void viewImage() {
145         startActivity(new Intent(this, ImageListView.class));
146     }
147 }
```

- Now add internet permission to your manifest. Your upload image will work at this point.
- Try running your app your image upload should work.

Uploading Image

- Now the lets move to the next part which is downloading uploaded images.

Showing Uploaded Images to a ListView in Android

> The process I will be following here is good when you have to load few images. Because here we will be downloading all the images at once. And if you have 100s or 1000s images to show up on ListView, you should not follow this. So to make your list ef cient check this tutorial here

Android Adding Items to List on Scroll

- Uncomment your line which were causing error before. Now to remove this error we need to create a new Activity.
- Create a blank activity. I just created ImageListView.
- Now when we will click the View Image button this activity should open.
- In this activity we will create a ListView. So come to the respective layout of this activity. (In my case it is **activity_image_list_view.xml**
- We need to create the following layout

Android Upload Image Using PHP MySQL

- You can use the following xml code for creating above layout

Android Download Image from Server

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
9
10
11      <ListView
12          android:layout_width="match_parent"
13          android:layout_height="wrap_content"
14          android:id="@+id/listView" />
15
16  </LinearLayout>
```

- We have to create a Custom List View for so we will create a new xml resource le for our Custom List View.
- I created image_list_view.xml

Image List View

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:orientation="horizontal"
4      android:layout_width="fill_parent"
5      android:layout_height="wrap_content">
6
7      <ImageView android:id="@+id/imageDownloaded"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:src="@drawable/abc_ic_menu_copy_mtrl_am_alpha"/>
11
12     <TextView android:id="@+id/textViewURL"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"/>
15
16 </LinearLayout>
```

- Now create a new class named CustomList for our Custom List View.

Custom List View Android Java

```
1  package net.simplifiedcoding.imageuploadsample;
2
```

```java
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

/**
 * Created by Belal on 7/22/2015.
 */
public class CustomList extends ArrayAdapter<String> {
    private String[] urls;
    private Bitmap[] bitmaps;
    private Activity context;

    public CustomList(Activity context, String[] urls, Bitmap[] bitmaps) {
        super(context, R.layout.image_list_view, urls);
        this.context = context;
        this.urls= urls;
        this.bitmaps= bitmaps;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = context.getLayoutInflater();
        View listViewItem = inflater.inflate(R.layout.image_list_view, null, true);
        TextView textViewURL = (TextView) listViewItem.findViewById(R.id.textViewURL);
        ImageView image = (ImageView) listViewItem.findViewById(R.id.imageDownloaded);

        textViewURL.setText(urls[position]);
        image.setImageBitmap(Bitmap.createScaledBitmap(bitmaps[position],100,50,false));
        return  listViewItem;
    }
}
```

- Now we need to fetch all the Bitmaps from server. So for this we will create a new class. I created GetAllImages.java

```
3  import android.app.ProgressDialog;
4  import android.graphics.Bitmap;
5  import android.graphics.BitmapFactory;
6  import android.os.AsyncTask;
7
8  import org.json.JSONArray;
9  import org.json.JSONException;
10 import org.json.JSONObject;
11
12 import java.io.IOException;
13 import java.net.MalformedURLException;
14 import java.net.URL;
15
16 /**
17  * Created by Belal on 9/19/2015.
18  */
19 public class GetAlImages {
20
21     public static String[] imageURLs;
22     public static Bitmap[] bitmaps;
23
24     public static final String JSON_ARRAY="result";
25     public static final String IMAGE_URL = "url";
26     private String json;
27     private JSONArray urls;
28
29     public GetAlImages(String json){
30         this.json = json;
31         try {
32             JSONObject jsonObject = new JSONObject(json);
33             urls = jsonObject.getJSONArray(JSON_ARRAY);
34         } catch (JSONException e) {
35             e.printStackTrace();
36         }
37     }
38
39     private Bitmap getImage(JSONObject jo){
40         URL url = null;
41         Bitmap image = null;
```

```
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return image;
    }

    public void getAllImages() throws JSONException {
        bitmaps = new Bitmap[urls.length()];

        imageURLs = new String[urls.length()];

        for(int i=0;i<urls.length();i++){
            imageURLs[i] = urls.getJSONObject(i).getString(IMAGE_URL);
            JSONObject jsonObject = urls.getJSONObject(i);
            bitmaps[i]=getImage(jsonObject);
        }
    }
}
```

- We will pass the json string having all the urls of our images to the constructor of this class.
- We will get the json string having all the urls from our getAllImages.php script.
- Now come to ImageListView.java class and write the following code

Android Get Images to ListView — Java

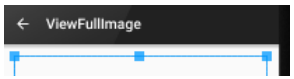
```
package net.simplifiedcoding.imageuploadsample;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.media.Image;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

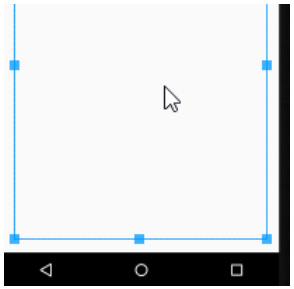
```
13 import android.widget.AdapterView;
14 import android.widget.Button;
15 import android.widget.ListView;
16 import android.widget.Toast;
17
18 import org.json.JSONException;
19
20 import java.io.BufferedReader;
21 import java.io.InputStreamReader;
22 import java.net.HttpURLConnection;
23 import java.net.URL;
24
25 public class ImageListView extends AppCompatActivity implements AdapterView.OnItemClickListene
26
27     private ListView listView;
28
29     public static final String GET_IMAGE_URL="http://simplifiedcoding.16mb.com/PhotoUpload/getAl
30
31     public GetAlImages getAlImages;
32
33     public static final String BITMAP_ID = "BITMAP_ID";
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         setContentView(R.layout.activity_image_list_view);
39
40         listView = (ListView) findViewById(R.id.listView);
41         listView.setOnItemClickListener(this);
42         getURLs();
43     }
44
45     private void getImages(){
46         class GetImages extends AsyncTask<Void,Void,Void>{
47             ProgressDialog loading;
48             @Override
49             protected void onPreExecute() {
50                 super.onPreExecute();
51                 loading = ProgressDialog.show(ImageListView.this,"Downloading images...","Please
```

```
55              protected void onPostExecute(Void v) {
56                  super.onPostExecute(v);
57                  loading.dismiss();
58                  //Toast.makeText(ImageListView.this,"Success",Toast.LENGTH_LONG).show();
59                  CustomList customList = new CustomList(ImageListView.this,GetAlImages.imageURLs,
60                  listView.setAdapter(customList);
61              }
62
63              @Override
64              protected Void doInBackground(Void... voids) {
65                  try {
66                      getAlImages.getAllImages();
67
68                  } catch (JSONException e) {
69                      e.printStackTrace();
70                  }
71                  return null;
72              }
73          }
74          GetImages getImages = new GetImages();
75          getImages.execute();
76      }
77
78      private void getURLs() {
79          class GetURLs extends AsyncTask<String,Void,String>{
80              ProgressDialog loading;
81
82              @Override
83              protected void onPreExecute() {
84                  super.onPreExecute();
85                  loading = ProgressDialog.show(ImageListView.this,"Loading...","Please Wait...",t
86              }
87
88              @Override
89              protected void onPostExecute(String s) {
90                  super.onPostExecute(s);
91                  loading.dismiss();
92                  getAlImages = new GetAlImages(s);
93                  getImages();
```

```
97              protected String doInBackground(String... strings) {
98                  BufferedReader bufferedReader = null;
99                  try {
100                     URL url = new URL(strings[0]);
101                     HttpURLConnection con = (HttpURLConnection) url.openConnection();
102                     StringBuilder sb = new StringBuilder();
103
104                     bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream
105
106                     String json;
107                     while((json = bufferedReader.readLine())!= null){
108                         sb.append(json+"\n");
109                     }
110
111                     return sb.toString().trim();
112
113                 }catch(Exception e){
114                     return null;
115                 }
116             }
117         }
118         GetURLs gu = new GetURLs();
119         gu.execute(GET_IMAGE_URL);
120     }
121
122     @Override
123     public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
124         Intent intent = new Intent(this, ViewFullImage.class);
125         intent.putExtra(BITMAP_ID,i);
126         startActivity(intent);
127     }
128 }
```

- Now we will create a new activity to see the full size image. I created ViewFullImage.java.
- We will create a ImageView to the layout le of this activity.





Android Fetch Image from Server

- The xml code for the above layout is

Android Fetch Image from Server

```
<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" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="net.simplifiedcoding.imageuploadsample.ViewFullImage">


    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageViewFull"
         />
</RelativeLayout>
```

- Inside your java class for this layout write the following code

Android Fetch Image from Server Java

```
package net.simplifiedcoding.imageuploadsample;

import android.content.Intent;
```

```
import android.view.MenuItem;
import android.widget.ImageView;

public class ViewFullImage extends AppCompatActivity {
    private ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_full_image);
        Intent intent = getIntent();
        int i = intent.getIntExtra(ImageListView.BITMAP_ID,0);

        imageView = (ImageView) findViewById(R.id.imageViewFull);
        imageView.setImageBitmap(GetAlImages.bitmaps[i]);

    }
}
```

- Now thats it. If you want the source code of this project you can download from here

Download Source

So thats it for this Android Upload Image Using PHP MySQL Tutorial friends. If you are having confusions please leave you comments below.