**Connecting Android App to Remote Database using PHP and MySQL**

You may have come across applications in which sends and receives data from the server. The server will store and fetch the data in a remote database. In this tutorial, I will explain how you can store and retrieve data from a remote MySQL database using PHP and how can you access the data in the Android Application. This tutorial will help you in learning the following:

* Adding data to a remote database in Android.
* Updating data in a remote database in Android.
* Fetching data from a remote database in Android.
* Deleting records in a remote database in Android.

First of all, we will see how can we set up the server and database on your desktop and then we will see how to access the data from the server.

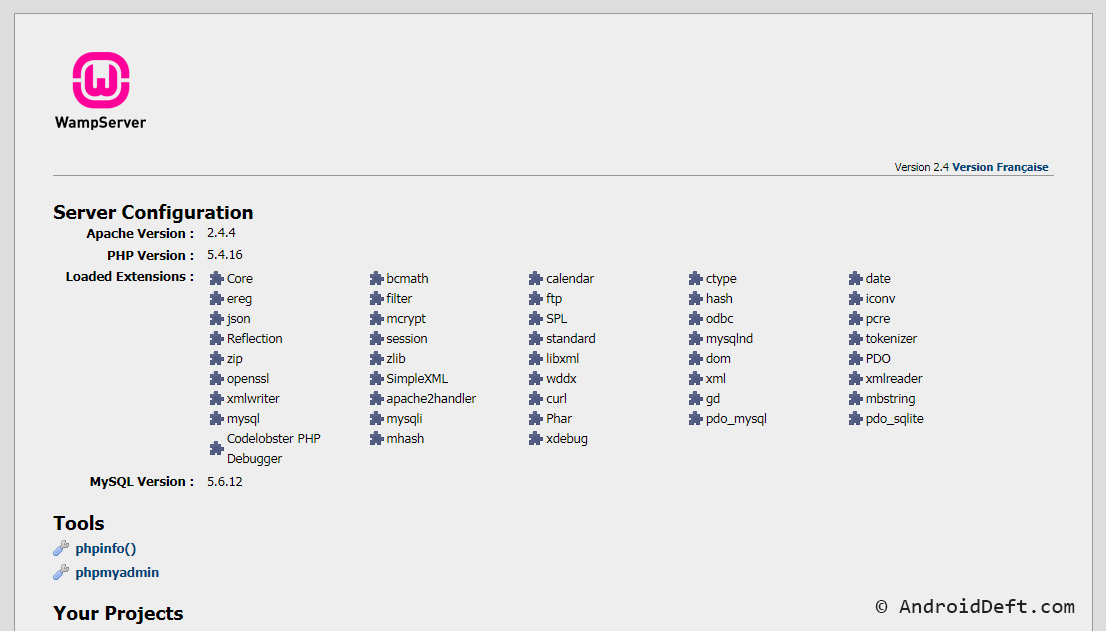
**Table of Contents**

* [1 Installing WAMP Server](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Installing_WAMP_Server)
* [2 Creating database in MySQL](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Creating_database_in_MySQL)
* [3 Connecting to MySQL database from PHP](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Connecting_to_MySQL_database_from_PHP)
* [4 MySQL CRUD Operations using PHP](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#MySQL_CRUD_Operations_using_PHP)
  + [4.1 Adding a new Movie (Inserting a row to the database)](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Adding_a_new_Movie_Inserting_a_row_to_the_database)
  + [4.2 Fetching All Movies (Retrieving data from database)](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Fetching_All_Movies_Retrieving_data_from_database)
  + [4.3 Fetching a single movie detail (Retrieving a single row from database)](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Fetching_a_single_movie_detail_Retrieving_a_single_row_from_database)
  + [4.4 Updating a movie (Updating a row in database)](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Updating_a_movie_Updating_a_row_in_database)
  + [4.5 Deleting a movie (Deleting a row from the database)](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Deleting_a_movie_Deleting_a_row_from_the_database)
* [5 Creating Android Project](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Creating_Android_Project)
* [6 Demonstration](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Demonstration)
* [7 Source Code and APK file](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Source_Code_and_APK_file)
* [8 Testing on Android Device](http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/#Testing_on_Android_Device)

**Installing WAMP Server**

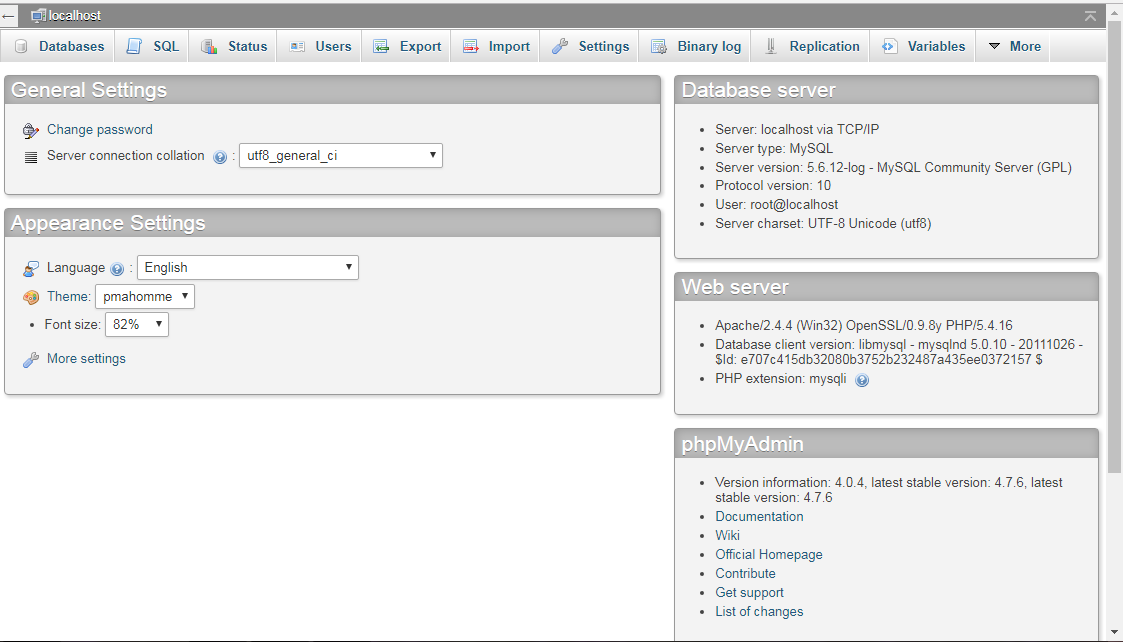
In order to create a web server on your local machine, you need to install [WAMP server](http://www.wampserver.com/en/#download-wrapper) or [XAMPP server](https://www.apachefriends.org) or any other similar software. The installation process is simple: You just need to download the software and install the application. I personally use [WAMP server](http://www.wampserver.com/en/#download-wrapper) and will be referring to it in the rest of the article.

Once you have installed wamp server, launch the program from Start => All Programs => WampServer => Start WampServer. You can verify whether the server has successfully started by launching <http://localhost/> in your browser. You should see a screen as shown below:

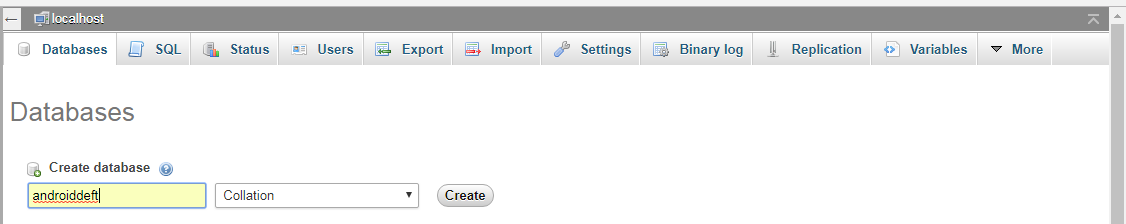


**Creating database in MySQL**

**1.** Open <http://localhost/phpmyadmin/> in your browser. This will prompt you for inputting username and password. By default username is *root*and there is no password (leave the password field empty). Once you login you will see the screen as shown below:



**2.** Click on **Databases** and create new database say *androiddeft.*



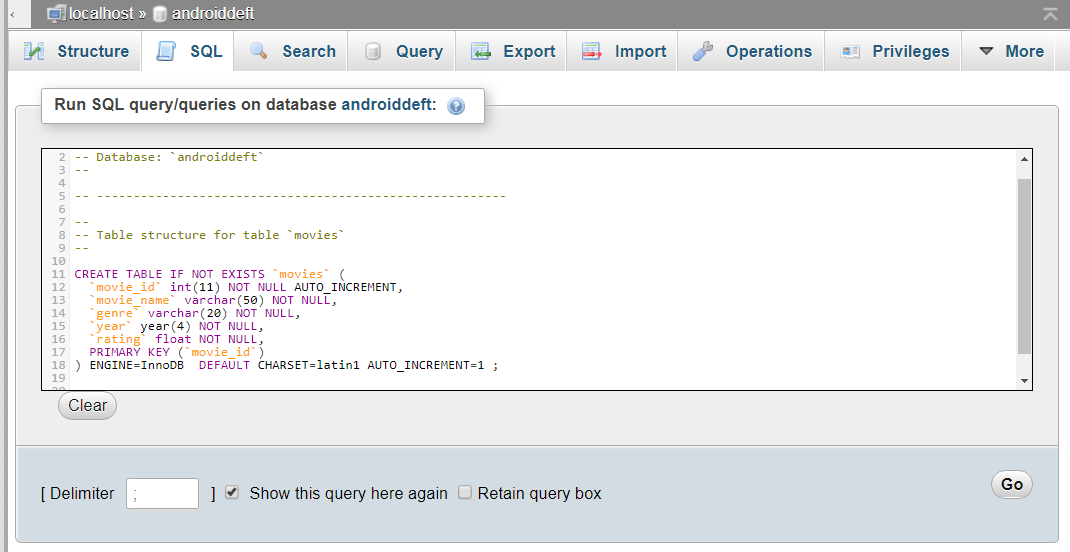
**3.** In order to store data in the database, you need to create tables inside the database. To achieve this, click on **SQL** tab and Copy & Paste the following query and click on **Go**.

movies.sql

MySQL



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | --  -- Database: `androiddeft`  --    -- --------------------------------------------------------    --  -- Table structure for table `movies`  --    CREATE TABLE IF NOT EXISTS `movies` (    `movie\_id` int(11) NOT NULL AUTO\_INCREMENT,    `movie\_name` varchar(50) NOT NULL,    `genre` varchar(20) NOT NULL,    `year` year(4) NOT NULL,    `rating` float NOT NULL,    PRIMARY KEY (`movie\_id`)  ) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;    -- --------------------------------------------------------  -- |



The above query creates **movies**table, used for storing movie details. We will be demonstrating the similar application used for demonstrating [SQLite database in Android](http://www.androiddeft.com/2017/05/06/android-storing-data-using-sqlite-database/).

**Connecting to MySQL database from PHP**

**1.** Navigate to the **www**folder of your WAMP installation (Usually located in **C:\wamp\www**, if you haven’t changed during the installation). This is the folder where you PHP projects reside.

**2.** Create a folder named **movies**inside **www** directory and a subdirectory, **db**inside it.

**3.** Create a file named *db\_connect.php*inside **db**folder and add the following code. This file contains database connection details as shown below and helps in connecting to MySQL database. This file has to be included in all files where we need to perform a database operation.

db\_connect.php

PHP



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <?php  define('DB\_USER', "root"); // db user  define('DB\_PASSWORD', ""); // db password (mention your db password here)  define('DB\_DATABASE', "androiddeft"); // database name  define('DB\_SERVER', "localhost"); // db server    $con = mysqli\_connect(DB\_SERVER,DB\_USER,DB\_PASSWORD,DB\_DATABASE);    // Check connection  if(mysqli\_connect\_errno()){  echo "Failed to connect to MySQL: " . mysqli\_connect\_error();  }  ?> |

**MySQL CRUD Operations using PHP**

We will see how to perform the CRUD (Create-Retrieve-Update-Delete) operations in PHP.

**Adding a new Movie (Inserting a row to the database)**

Create a file **add\_movie.php**inside **movies**folder and place the following code in it. This code helps in adding a new movie to the database. Here we check for mandatory parameters and prepare an insert query. The parameters need to be sent as POST parameter. If the movie got added successfully, then we display a success response. If there are any errors, then an error message is displayed.

add\_movie.php

PHP



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | <?php  include 'db/db\_connect.php';  $response = array();    //Check for mandatory parameters  if(isset($\_POST['movie\_name'])&&isset($\_POST['genre'])&&isset($\_POST['year'])&&isset($\_POST['rating'])){  $movieName = $\_POST['movie\_name'];  $genre = $\_POST['genre'];  $year = $\_POST['year'];  $rating = $\_POST['rating'];  //Query to insert a movie  $query = "INSERT INTO movies( movie\_name, genre, year, rating) VALUES (?,?,?,?)";  //Prepare the query  if($stmt = $con->prepare($query)){  //Bind parameters  $stmt->bind\_param("ssis",$movieName,$genre,$year,$rating);  //Exceting MySQL statement  $stmt->execute();  //Check if data got inserted  if($stmt->affected\_rows == 1){  $response["success"] = 1;  $response["message"] = "Movie Successfully Added";  }else{  //Some error while inserting  $response["success"] = 0;  $response["message"] = "Error while adding movie";  }  }else{  //Some error while inserting  $response["success"] = 0;  $response["message"] = mysqli\_error($con);  }    }else{  //Mandatory parameters are missing  $response["success"] = 0;  $response["message"] = "missing mandatory parameters";  }  //Displaying JSON response  echo json\_encode($response);  ?> |

When a row gets inserted successfully, we get the following response:



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 1,    "message": "Movie Successfully Added"  } |

If there are any parameters missing in the request then we display following response:



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 0,    "message": "missing mandatory parameters"  } |

If any error occurs then,



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 0,    "message": "Error while adding movie"  } |

**Fetching All Movies (Retrieving data from database)**

Create file **fetch\_all\_movies.php**inside the PHP project and add the following code for fetching all movie details:

fetch\_all\_movies.php

PHP



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | <?php  include 'db/db\_connect.php';  //Query to select movie id and movie name  $query = "SELECT movie\_id, movie\_name FROM movies";  $result = array();  $movieArray = array();  $response = array();  //Prepare the query  if($stmt = $con->prepare($query)){  $stmt->execute();  //Bind the fetched data to $movieId and $movieName  $stmt->bind\_result($movieId,$movieName);  //Fetch 1 row at a time  while($stmt->fetch()){  //Populate the movie array  $movieArray["movie\_id"] = $movieId;  $movieArray["movie\_name"] = $movieName;  $result[]=$movieArray;  }  $stmt->close();  $response["success"] = 1;  $response["data"] = $result;    }else{  //Some error while fetching data  $response["success"] = 0;  $response["message"] = mysqli\_error($con);  }  //Display JSON response  echo json\_encode($response);    ?> |

Here we query the database for *movie\_id* and *movie\_name*and we iterate over the fetched result and populate the result array.

On successful fetching of result we get the following response:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | {    "success": 1,    "data": [      {        "movie\_id": 1,        "movie\_name": "Raees"      },      {        "movie\_id": 3,        "movie\_name": "Dangal"      }    ]  } |

**Fetching a single movie detail (Retrieving a single row from database)**

We can fetch the details of a single movie by passing the *movie\_id*in the request. That is by using the primary key of the table, we can fetch a unique row from the database.

Create a file **get\_movie\_details.php**inside the **movies** folder and add the following code. The code expects the mandatory parameter *movie\_id*to be passed in an HTTP GET method.

get\_movie\_details.php

PHP



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | <?php  include 'db/db\_connect.php';  $movieArray = array();  $response = array();  //Check for mandatory parameter movie\_id  if(isset($\_GET['movie\_id'])){  $movieId = $\_GET['movie\_id'];  //Query to fetch movie details  $query = "SELECT movie\_name, genre, year, rating FROM movies WHERE movie\_id=?";  if($stmt = $con->prepare($query)){  //Bind movie\_id parameter to the query  $stmt->bind\_param("i",$movieId);  $stmt->execute();  //Bind fetched result to variables $movieName, $genre, $year and $rating  $stmt->bind\_result($movieName,$genre,$year,$rating);  //Check for results  if($stmt->fetch()){  //Populate the movie array  $movieArray["movie\_id"] = $movieId;  $movieArray["movie\_name"] = $movieName;  $movieArray["genre"] = $genre;  $movieArray["year"] = $year;  $movieArray["rating"] = round($rating,1);  $response["success"] = 1;  $response["data"] = $movieArray;  }else{  //When movie is not found  $response["success"] = 0;  $response["message"] = "Movie not found";  }  $stmt->close();      }else{  //Whe some error occurs  $response["success"] = 0;  $response["message"] = mysqli\_error($con);  }    }else{  //When the mandatory parameter movie\_id is missing  $response["success"] = 0;  $response["message"] = "missing parameter movie\_id";  }  //Display JSON response  echo json\_encode($response);  ?> |

Now if you hit <http://localhost/movies/get_movie_details.php?movie_id=1> in your browser you should be able to see the following response:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | {    "success": 1,    "data": {      "movie\_id": "1",      "movie\_name": "Raees",      "genre": "Crime/Thriller",      "year": 2017,      "rating": 7.2    }  } |

When a movie is not found, you will get:



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 0,    "message": "Movie not found"  } |

**Updating a movie (Updating a row in database)**

When you wish to update a row in the database, you send the column value that needs to be updated along with the primary key of the table.

Create a file named **update\_movie.php**and add the following code:

update\_movie.php

PHP



|  |
| --- |
| <?php  include 'db/db\_connect.php';  $response = array();    //Check for mandatory parameters  if(isset($\_POST['movie\_name'])&&isset($\_POST['movie\_id'])&&isset($\_POST['genre'])&&isset($\_POST['year'])&&isset($\_POST['rating'])){  $movieName = $\_POST['movie\_name'];  $movieId = $\_POST['movie\_id'];  $genre = $\_POST['genre'];  $year = $\_POST['year'];  $rating = $\_POST['rating'];  //Query to update a movie  $query = "UPDATE movies SET movie\_name=?,genre=?,year=?,rating=? WHERE movie\_id=?";  //Prepare the query  if($stmt = $con->prepare($query)){  //Bind parameters  $stmt->bind\_param("ssisi",$movieName,$genre,$year,$rating,$movieId);  //Exceting MySQL statement  $stmt->execute();  //Check if data got updated  if($stmt->affected\_rows == 1){  $response["success"] = 1;  $response["message"] = "Movie successfully updated";  }else{  //When movie is not found  $response["success"] = 0;  $response["message"] = "Movie not found";  }  }else{  //Some error while updating  $response["success"] = 0;  $response["message"] = mysqli\_error($con);  }    }else{  //Mandatory parameters are missing  $response["success"] = 0;  $response["message"] = "missing mandatory parameters";  }  //Displaying JSON response  echo json\_encode($response);  ?> |

On successful update you will get the following response;



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 1,    "message": "Movie successfully updated"  } |

If the movie is not found,



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 0,    "message": "MovieMovie not found"  } |

**Deleting a movie (Deleting a row from the database)**

When you want to delete a movie, you need to pass the *movie\_id*in the request.

Create a file **delete\_movie.php**and add the following code:

delete\_movie.php

PHP



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <?php  include 'db/db\_connect.php';  $response = array();  //Check for mandatory parameter movie\_id  if(isset($\_POST['movie\_id'])){  $movieId = $\_POST['movie\_id'];  $query = "DELETE FROM movies WHERE movie\_id=?";  if($stmt = $con->prepare($query)){  //Bind movie\_id parameter to the query  $stmt->bind\_param("i",$movieId);  $stmt->execute();  //Check if the movie got deleted  if($stmt->affected\_rows == 1){  $response["success"] = 1;  $response["message"] = "Movie got deleted successfully";  }else{  $response["success"] = 0;  $response["message"] = "Movie not found";  }  }else{  $response["success"] = 0;  $response["message"] = mysqli\_error($con);  }    }else{  $response["success"] = 0;  $response["message"] = "missing parameter movie\_id";  }  echo json\_encode($response);  ?> |

Here the code requires *movie\_id*to be passed in an HTPP POST method. Once the movie gets deleted, you will get the following response:



|  |  |
| --- | --- |
| 1  2  3  4 | {    "success": 1,    "message": "Movie got deleted successfully"  } |

**Creating Android Project**

Now we will see how to use the created APIs in the Android application.

**1.** Create a new project in Android Studio with the name **Remote MySQL Connection**.

**2.**  Add Internet and Check Network Status permissions in **AndroidManifest.xml**

AndroidManifest.xml

XHTML



|  |  |
| --- | --- |
| 5  6 | <uses-permission android:name="android.permission.INTERNET" />      <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" /> |

**3.** Add the string variables to be used in this application to **strings.xml**

strings.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <resources>      <string name="app\_name">Remote MySQL Connection</string>      <string name="view\_all">View All Movies</string>      <string name="add\_new">Add New Movie</string>      <string name="title\_activity\_movie\_update\_delete">MovieUpdateDeleteActivity</string>      <string name="movieName">Movie Name</string>      <string name="genre">Genre</string>      <string name="year">Year</string>      <string name="rating">Rating</string>      <string name="update">Update</string>      <string name="delete">Delete</string>      <string name="add">Add</string>  </resources> |

**4.** Create a package named **helper** inside the main package and add a class named **CheckNetworkStatus.java.**This class has a static method *isNetworkAvailable,*which checks the network status of the device.

CheckNetworkStatus.java

Java



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | package com.androiddeft.remotemysqlconnection.helper;    import android.content.Context;  import android.net.ConnectivityManager;  import android.net.NetworkInfo;    /\*\*  \* Created by Abhi on 23 Jun 2017 023.  \* Checks whether the device is connected to internet or not  \*/    public class CheckNetworkStatus {        public static boolean isNetworkAvailable(Context context) {          ConnectivityManager connectivityManager                  = (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY\_SERVICE);          NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();          return activeNetworkInfo != null && activeNetworkInfo.isConnected();      }  } |

**5.** Create a class **HttpJsonParser i**nside the **helper**package. This class helper has a method *makeHttpRequest*which helps in making HTTP GET and POST requests. We are reusing this class from [JSON Parsing Tutorial](http://www.androiddeft.com/2017/09/10/json-parsing-android-http/).

HttpJsonParser.java

Java



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95 | package com.androiddeft.remotemysqlconnection.helper;    /\*\*  \* Created by Abhi on 19 Jun 2017 019.  \*/      import java.io.BufferedReader;  import java.io.IOException;  import java.io.InputStream;  import java.io.InputStreamReader;  import java.io.UnsupportedEncodingException;  import java.net.HttpURLConnection;  import java.net.ProtocolException;  import java.net.URL;  import java.util.Map;    import org.json.JSONException;  import org.json.JSONObject;    import android.net.Uri;  import android.util.Log;    public class HttpJsonParser {        static InputStream is = null;      static JSONObject jObj = null;      static String json = "";      HttpURLConnection urlConnection = null;        // function get json from url      // by making HTTP POST or GET method      public JSONObject makeHttpRequest(String url, String method,                                        Map<String, String> params) {            try {              Uri.Builder builder = new Uri.Builder();              URL urlObj;              String encodedParams = "";              if (params != null) {                  for (Map.Entry<String, String> entry : params.entrySet()) {                      builder.appendQueryParameter(entry.getKey(), entry.getValue());                  }              }              if (builder.build().getEncodedQuery() != null) {                  encodedParams =  builder.build().getEncodedQuery();                }              if ("GET".equals(method)) {                  url = url + "?" + encodedParams;                  urlObj = new URL(url);                  urlConnection = (HttpURLConnection) urlObj.openConnection();                  urlConnection.setRequestMethod(method);                  } else {                  urlObj = new URL(url);                  urlConnection = (HttpURLConnection) urlObj.openConnection();                  urlConnection.setRequestMethod(method);                  urlConnection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");                  urlConnection.setRequestProperty("Content-Length", String.valueOf(encodedParams.getBytes().length));                  urlConnection.getOutputStream().write(encodedParams.getBytes());              }                  urlConnection.connect();              is = urlConnection.getInputStream();              BufferedReader reader = new BufferedReader(new InputStreamReader(is));              StringBuilder sb = new StringBuilder();              String line;              while ((line = reader.readLine()) != null) {                  sb.append(line + "\n");              }              is.close();              json = sb.toString();              jObj = new JSONObject(json);              } catch (UnsupportedEncodingException e) {              e.printStackTrace();          } catch (ProtocolException e) {              e.printStackTrace();          } catch (IOException e) {              e.printStackTrace();          } catch (JSONException e) {              Log.e("JSON Parser", "Error parsing data " + e.toString());          } catch (Exception e) {              Log.e("Exception", "Error parsing data " + e.toString());          }            // return JSON String          return jObj;        }  } |

**6.** Add 2 buttons to **activity\_main.xml** (I have renamed it to **activity\_remote\_my\_sql.xml**) with labels **View All Movies** and **Add New Movie:**

activity\_remote\_my\_sql.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:id="@+id/activity\_remote\_my\_sql"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical"      android:paddingBottom="@dimen/activity\_vertical\_margin"      android:paddingLeft="@dimen/activity\_horizontal\_margin"      android:paddingRight="@dimen/activity\_horizontal\_margin"      android:paddingTop="@dimen/activity\_vertical\_margin"      tools:context="com.androiddeft.remotemysqlconnection.RemoteMySQLActivity">        <Button          android:id="@+id/viewAllBtn"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:text="@string/view\_all" />        <Button          android:id="@+id/addNewBtn"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:text="@string/add\_new" />  </LinearLayout> |

**7.** Create a file **list\_background.xml**inside **drawable** with the following code. This is used to style the list view, used to display the list of movies.

list\_background.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | <?xml version="1.0" encoding="utf-8"?>  <shape xmlns:android="http://schemas.android.com/apk/res/android"      android:shape="rectangle">        <!-- View background color -->      <solid          android:color="#fff">      </solid>        <!-- View border color and width -->      <stroke          android:width="0dp"          android:color="#aaa">      </stroke>        <!-- The radius makes the corners rounded -->      <corners          android:radius="2dp">      </corners>    </shape> |

**8.** Create a layout file **list\_item.xml** inside the **layout** folder. This layout file is used to display a custom ListView

list\_item.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:background="@drawable/list\_background"      android:elevation="1dp"      android:orientation="vertical">        <!-- Hidden TextView used to hold movieId -->      <TextView          android:id="@+id/movieId"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:visibility="gone" />        <TextView          android:id="@+id/movieName"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:padding="10dp"          android:textSize="18sp"          android:textStyle="bold" />  </RelativeLayout> |

**9.** Create a new Activity called **MovieListingActivity**with layout file **activity\_movie\_listing.xml**

The layout contains a ListView to display the movies list:

activity\_movie\_listing.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <?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:id="@+id/activity\_movie\_listing"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:background="#eee"      android:paddingBottom="@dimen/activity\_vertical\_margin"      android:paddingLeft="@dimen/activity\_horizontal\_margin"      android:paddingRight="@dimen/activity\_horizontal\_margin"      android:paddingTop="@dimen/activity\_vertical\_margin"      tools:context="com.androiddeft.remotemysqlconnection.MovieListingActivity">        <ListView          android:id="@+id/movieList"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:divider="@android:color/transparent"          android:dividerHeight="5dp">        </ListView>    </RelativeLayout> |

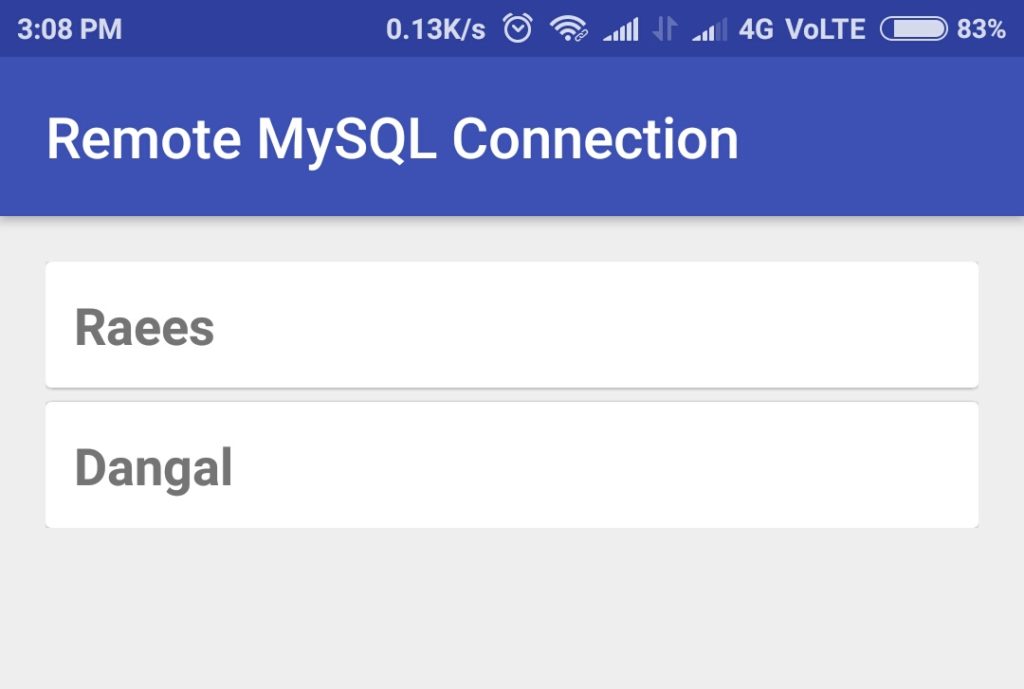
**MovieListingActivity**fetches the list of movies from the server using AsyncTask and populates the ListView

MovieListingActivity.java

Java



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149 | package com.androiddeft.remotemysqlconnection;    import android.app.ProgressDialog;  import android.content.Intent;  import android.os.AsyncTask;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.AdapterView;  import android.widget.ListAdapter;  import android.widget.ListView;  import android.widget.SimpleAdapter;  import android.widget.TextView;  import android.widget.Toast;    import com.androiddeft.remotemysqlconnection.helper.CheckNetworkStatus;  import com.androiddeft.remotemysqlconnection.helper.HttpJsonParser;    import org.json.JSONArray;  import org.json.JSONException;  import org.json.JSONObject;    import java.util.ArrayList;  import java.util.HashMap;    public class MovieListingActivity extends AppCompatActivity {      private static final String KEY\_SUCCESS = "success";      private static final String KEY\_DATA = "data";      private static final String KEY\_MOVIE\_ID = "movie\_id";      private static final String KEY\_MOVIE\_NAME = "movie\_name";      private static final String BASE\_URL = "http://192.168.43.72/movies/";      private ArrayList<HashMap<String, String>> movieList;      private ListView movieListView;      private ProgressDialog pDialog;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_movie\_listing);          movieListView = (ListView) findViewById(R.id.movieList);          new FetchMoviesAsyncTask().execute();        }        /\*\*       \* Fetches the list of movies from the server       \*/      private class FetchMoviesAsyncTask extends AsyncTask<String, String, String> {          @Override          protected void onPreExecute() {              super.onPreExecute();              //Display progress bar              pDialog = new ProgressDialog(MovieListingActivity.this);              pDialog.setMessage("Loading movies. Please wait...");              pDialog.setIndeterminate(false);              pDialog.setCancelable(false);              pDialog.show();          }            @Override          protected String doInBackground(String... params) {              HttpJsonParser httpJsonParser = new HttpJsonParser();              JSONObject jsonObject = httpJsonParser.makeHttpRequest(                      BASE\_URL + "fetch\_all\_movies.php", "GET", null);              try {                  int success = jsonObject.getInt(KEY\_SUCCESS);                  JSONArray movies;                  if (success == 1) {                      movieList = new ArrayList<>();                      movies = jsonObject.getJSONArray(KEY\_DATA);                      //Iterate through the response and populate movies list                      for (int i = 0; i < movies.length(); i++) {                          JSONObject movie = movies.getJSONObject(i);                          Integer movieId = movie.getInt(KEY\_MOVIE\_ID);                          String movieName = movie.getString(KEY\_MOVIE\_NAME);                          HashMap<String, String> map = new HashMap<String, String>();                          map.put(KEY\_MOVIE\_ID, movieId.toString());                          map.put(KEY\_MOVIE\_NAME, movieName);                          movieList.add(map);                      }                  }              } catch (JSONException e) {                  e.printStackTrace();              }              return null;          }            protected void onPostExecute(String result) {              pDialog.dismiss();              runOnUiThread(new Runnable() {                  public void run() {                      populateMovieList();                  }              });          }        }        /\*\*       \* Updating parsed JSON data into ListView       \* \*/      private void populateMovieList() {          ListAdapter adapter = new SimpleAdapter(                  MovieListingActivity.this, movieList,                  R.layout.list\_item, new String[]{KEY\_MOVIE\_ID,                  KEY\_MOVIE\_NAME},                  new int[]{R.id.movieId, R.id.movieName});          // updating listview          movieListView.setAdapter(adapter);          //Call MovieUpdateDeleteActivity when a movie is clicked          movieListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {              @Override              public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {                  //Check for network connectivity                  if (CheckNetworkStatus.isNetworkAvailable(getApplicationContext())) {                      String movieId = ((TextView) view.findViewById(R.id.movieId))                              .getText().toString();                      Intent intent = new Intent(getApplicationContext(),                              MovieUpdateDeleteActivity.class);                      intent.putExtra(KEY\_MOVIE\_ID, movieId);                      startActivityForResult(intent, 20);                    } else {                      Toast.makeText(MovieListingActivity.this,                              "Unable to connect to internet",                              Toast.LENGTH\_LONG).show();                    }                  }          });        }        @Override      protected void onActivityResult(int requestCode, int resultCode, Intent data) {          super.onActivityResult(requestCode, resultCode, data);          if (resultCode == 20) {              // If the result code is 20 that means that              // the user has deleted/updated the movie.              // So refresh the movie listing              Intent intent = getIntent();              finish();              startActivity(intent);          }      }    } |



**10.** Create an activity **MovieUpdateDeleteActivity** with layout file **activity\_movie\_update\_delete.xml**

MovieUpdateDeleteActivity.java

Java



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313 | package com.androiddeft.remotemysqlconnection;    import android.app.ProgressDialog;  import android.content.DialogInterface;  import android.content.Intent;  import android.os.AsyncTask;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;  import android.app.AlertDialog;    import com.androiddeft.remotemysqlconnection.helper.CheckNetworkStatus;  import com.androiddeft.remotemysqlconnection.helper.HttpJsonParser;    import org.json.JSONException;  import org.json.JSONObject;    import java.util.HashMap;  import java.util.Map;    public class MovieUpdateDeleteActivity extends AppCompatActivity {      private static String STRING\_EMPTY = "";      private static final String KEY\_SUCCESS = "success";      private static final String KEY\_DATA = "data";      private static final String KEY\_MOVIE\_ID = "movie\_id";      private static final String KEY\_MOVIE\_NAME = "movie\_name";      private static final String KEY\_GENRE = "genre";      private static final String KEY\_YEAR = "year";      private static final String KEY\_RATING = "rating";      private static final String BASE\_URL = "http://192.168.43.72/movies/";      private String movieId;      private EditText movieNameEditText;      private EditText genreEditText;      private EditText yearEditText;      private EditText ratingEditText;      private String movieName;      private String genre;      private String year;      private String rating;      private Button deleteButton;      private Button updateButton;      private int success;      private ProgressDialog pDialog;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_movie\_update\_delete);          Intent intent = getIntent();          movieNameEditText = (EditText) findViewById(R.id.txtMovieNameUpdate);          genreEditText = (EditText) findViewById(R.id.txtGenreUpdate);          yearEditText = (EditText) findViewById(R.id.txtYearUpdate);          ratingEditText = (EditText) findViewById(R.id.txtRatingUpdate);            movieId = intent.getStringExtra(KEY\_MOVIE\_ID);          new FetchMovieDetailsAsyncTask().execute();          deleteButton = (Button) findViewById(R.id.btnDelete);          deleteButton.setOnClickListener(new View.OnClickListener() {              @Override              public void onClick(View view) {                  confirmDelete();              }          });          updateButton = (Button) findViewById(R.id.btnUpdate);          updateButton.setOnClickListener(new View.OnClickListener() {              @Override              public void onClick(View view) {                  if (CheckNetworkStatus.isNetworkAvailable(getApplicationContext())) {                      updateMovie();                    } else {                      Toast.makeText(MovieUpdateDeleteActivity.this,                              "Unable to connect to internet",                              Toast.LENGTH\_LONG).show();                    }                }          });          }        /\*\*       \* Fetches single movie details from the server       \*/      private class FetchMovieDetailsAsyncTask extends AsyncTask<String, String, String> {          @Override          protected void onPreExecute() {              super.onPreExecute();              //Display progress bar              pDialog = new ProgressDialog(MovieUpdateDeleteActivity.this);              pDialog.setMessage("Loading Movie Details. Please wait...");              pDialog.setIndeterminate(false);              pDialog.setCancelable(false);              pDialog.show();          }            @Override          protected String doInBackground(String... params) {              HttpJsonParser httpJsonParser = new HttpJsonParser();              Map<String, String> httpParams = new HashMap<>();              httpParams.put(KEY\_MOVIE\_ID, movieId);              JSONObject jsonObject = httpJsonParser.makeHttpRequest(                      BASE\_URL + "get\_movie\_details.php", "GET", httpParams);              try {                  int success = jsonObject.getInt(KEY\_SUCCESS);                  JSONObject movie;                  if (success == 1) {                      //Parse the JSON response                      movie = jsonObject.getJSONObject(KEY\_DATA);                      movieName = movie.getString(KEY\_MOVIE\_NAME);                      genre = movie.getString(KEY\_GENRE);                      year = movie.getString(KEY\_YEAR);                      rating = movie.getString(KEY\_RATING);                    }              } catch (JSONException e) {                  e.printStackTrace();              }              return null;          }            protected void onPostExecute(String result) {              pDialog.dismiss();              runOnUiThread(new Runnable() {                  public void run() {                      //Populate the Edit Texts once the network activity is finished executing                      movieNameEditText.setText(movieName);                      genreEditText.setText(genre);                      yearEditText.setText(year);                      ratingEditText.setText(rating);                    }              });          }          }        /\*\*       \* Displays an alert dialogue to confirm the deletion       \*/      private void confirmDelete() {          AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(                  MovieUpdateDeleteActivity.this);          alertDialogBuilder.setMessage("Are you sure, you want to delete this movie?");          alertDialogBuilder.setPositiveButton("Delete",                  new DialogInterface.OnClickListener() {                      @Override                      public void onClick(DialogInterface arg0, int arg1) {                          if (CheckNetworkStatus.isNetworkAvailable(getApplicationContext())) {                              //If the user confirms deletion, execute DeleteMovieAsyncTask                              new DeleteMovieAsyncTask().execute();                          } else {                              Toast.makeText(MovieUpdateDeleteActivity.this,                                      "Unable to connect to internet",                                      Toast.LENGTH\_LONG).show();                            }                      }                  });            alertDialogBuilder.setNegativeButton("Cancel", null);            AlertDialog alertDialog = alertDialogBuilder.create();          alertDialog.show();      }        /\*\*       \* AsyncTask to delete a movie       \*/      private class DeleteMovieAsyncTask extends AsyncTask<String, String, String> {          @Override          protected void onPreExecute() {              super.onPreExecute();              //Display progress bar              pDialog = new ProgressDialog(MovieUpdateDeleteActivity.this);              pDialog.setMessage("Deleting Movie. Please wait...");              pDialog.setIndeterminate(false);              pDialog.setCancelable(false);              pDialog.show();          }            @Override          protected String doInBackground(String... params) {              HttpJsonParser httpJsonParser = new HttpJsonParser();              Map<String, String> httpParams = new HashMap<>();              //Set movie\_id parameter in request              httpParams.put(KEY\_MOVIE\_ID, movieId);              JSONObject jsonObject = httpJsonParser.makeHttpRequest(                      BASE\_URL + "delete\_movie.php", "POST", httpParams);              try {                  success = jsonObject.getInt(KEY\_SUCCESS);              } catch (JSONException e) {                  e.printStackTrace();              }              return null;          }            protected void onPostExecute(String result) {              pDialog.dismiss();              runOnUiThread(new Runnable() {                  public void run() {                      if (success == 1) {                          //Display success message                          Toast.makeText(MovieUpdateDeleteActivity.this,                                  "Movie Deleted", Toast.LENGTH\_LONG).show();                          Intent i = getIntent();                          //send result code 20 to notify about movie deletion                          setResult(20, i);                          finish();                        } else {                          Toast.makeText(MovieUpdateDeleteActivity.this,                                  "Some error occurred while deleting movie",                                  Toast.LENGTH\_LONG).show();                        }                  }              });          }      }        /\*\*       \* Checks whether all files are filled. If so then calls UpdateMovieAsyncTask.       \* Otherwise displays Toast message informing one or more fields left empty       \*/      private void updateMovie() {              if (!STRING\_EMPTY.equals(movieNameEditText.getText().toString()) &&                  !STRING\_EMPTY.equals(genreEditText.getText().toString()) &&                  !STRING\_EMPTY.equals(yearEditText.getText().toString()) &&                  !STRING\_EMPTY.equals(ratingEditText.getText().toString())) {                movieName = movieNameEditText.getText().toString();              genre = genreEditText.getText().toString();              year = yearEditText.getText().toString();              rating = ratingEditText.getText().toString();              new UpdateMovieAsyncTask().execute();          } else {              Toast.makeText(MovieUpdateDeleteActivity.this,                      "One or more fields left empty!",                      Toast.LENGTH\_LONG).show();            }          }      /\*\*       \* AsyncTask for updating a movie details       \*/        private class UpdateMovieAsyncTask extends AsyncTask<String, String, String> {          @Override          protected void onPreExecute() {              super.onPreExecute();              //Display progress bar              pDialog = new ProgressDialog(MovieUpdateDeleteActivity.this);              pDialog.setMessage("Updating Movie. Please wait...");              pDialog.setIndeterminate(false);              pDialog.setCancelable(false);              pDialog.show();          }            @Override          protected String doInBackground(String... params) {              HttpJsonParser httpJsonParser = new HttpJsonParser();              Map<String, String> httpParams = new HashMap<>();              //Populating request parameters              httpParams.put(KEY\_MOVIE\_ID, movieId);              httpParams.put(KEY\_MOVIE\_NAME, movieName);              httpParams.put(KEY\_GENRE, genre);              httpParams.put(KEY\_YEAR, year);              httpParams.put(KEY\_RATING, rating);              JSONObject jsonObject = httpJsonParser.makeHttpRequest(                      BASE\_URL + "update\_movie.php", "POST", httpParams);              try {                  success = jsonObject.getInt(KEY\_SUCCESS);              } catch (JSONException e) {                  e.printStackTrace();              }              return null;          }            protected void onPostExecute(String result) {              pDialog.dismiss();              runOnUiThread(new Runnable() {                  public void run() {                      if (success == 1) {                          //Display success message                          Toast.makeText(MovieUpdateDeleteActivity.this,                                  "Movie Updated", Toast.LENGTH\_LONG).show();                          Intent i = getIntent();                          //send result code 20 to notify about movie update                          setResult(20, i);                          finish();                        } else {                          Toast.makeText(MovieUpdateDeleteActivity.this,                                  "Some error occurred while updating movie",                                  Toast.LENGTH\_LONG).show();                        }                  }              });          }      }  } |

This activity has EditTexts, which is used both to display and edit the movie details. It has 2 buttons, 1 to update the movie details and another to delete the movie

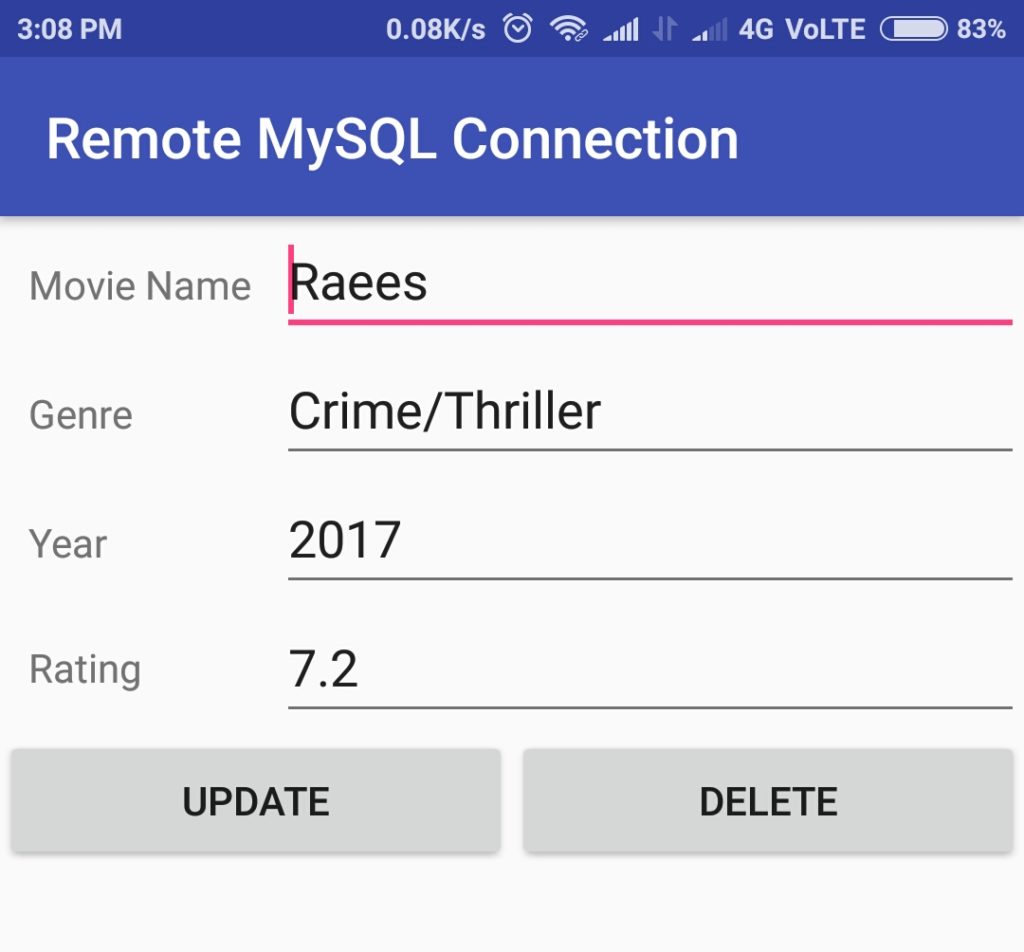
activity\_movie\_update\_delete.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117 | <?xml version="1.0" encoding="utf-8"?>  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="wrap\_content">        <LinearLayout          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:orientation="vertical">            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.25"                  android:text="@string/movieName" />                <EditText                  android:layout\_width="0dp"                  android:id="@+id/txtMovieNameUpdate"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10" />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.25"                  android:text="@string/genre" />                <EditText                  android:layout\_width="0dp"                  android:id="@+id/txtGenreUpdate"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10" />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.25"                  android:text="@string/year" />                <EditText                  android:id="@+id/txtYearUpdate"                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10" />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_gravity="center\_vertical"                  android:layout\_weight="0.25"                    android:text="@string/rating" />                  <EditText                  android:id="@+id/txtRatingUpdate"                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10"                    />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal">                  <Button                  android:id="@+id/btnUpdate"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.50"                  android:text="@string/update" />              <Button                  android:id="@+id/btnDelete"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.50"                  android:text="@string/delete" />          </LinearLayout>        </LinearLayout>  </ScrollView> |

Below is the snapshot of the activity:



**11.** Create another activity named **AddMovieActivity**with layout file **activity\_add\_movie.xml**

AddMovieActivity.java

Java



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151 | package com.androiddeft.remotemysqlconnection;    import android.app.ProgressDialog;  import android.content.Intent;  import android.os.AsyncTask;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;    import com.androiddeft.remotemysqlconnection.helper.CheckNetworkStatus;  import com.androiddeft.remotemysqlconnection.helper.HttpJsonParser;    import org.json.JSONException;  import org.json.JSONObject;    import java.util.HashMap;  import java.util.Map;    public class AddMovieActivity extends AppCompatActivity {      private static final String KEY\_SUCCESS = "success";      private static final String KEY\_MOVIE\_NAME = "movie\_name";      private static final String KEY\_GENRE = "genre";      private static final String KEY\_YEAR = "year";      private static final String KEY\_RATING = "rating";      private static final String BASE\_URL = "http://192.168.43.72/movies/";      private static String STRING\_EMPTY = "";      private EditText movieNameEditText;      private EditText genreEditText;      private EditText yearEditText;      private EditText ratingEditText;      private String movieName;      private String genre;      private String year;      private String rating;      private Button addButton;      private int success;      private ProgressDialog pDialog;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_add\_movie);          movieNameEditText = (EditText) findViewById(R.id.txtMovieNameAdd);          genreEditText = (EditText) findViewById(R.id.txtGenreAdd);          yearEditText = (EditText) findViewById(R.id.txtYearAdd);          ratingEditText = (EditText) findViewById(R.id.txtRatingAdd);          addButton = (Button) findViewById(R.id.btnAdd);          addButton.setOnClickListener(new View.OnClickListener() {              @Override              public void onClick(View view) {                  if (CheckNetworkStatus.isNetworkAvailable(getApplicationContext())) {                      addMovie();                  } else {                      Toast.makeText(AddMovieActivity.this,                              "Unable to connect to internet",                              Toast.LENGTH\_LONG).show();                    }                }          });        }        /\*\*       \* Checks whether all files are filled. If so then calls AddMovieAsyncTask.       \* Otherwise displays Toast message informing one or more fields left empty       \*/      private void addMovie() {          if (!STRING\_EMPTY.equals(movieNameEditText.getText().toString()) &&                  !STRING\_EMPTY.equals(genreEditText.getText().toString()) &&                  !STRING\_EMPTY.equals(yearEditText.getText().toString()) &&                  !STRING\_EMPTY.equals(ratingEditText.getText().toString())) {                movieName = movieNameEditText.getText().toString();              genre = genreEditText.getText().toString();              year = yearEditText.getText().toString();              rating = ratingEditText.getText().toString();              new AddMovieAsyncTask().execute();          } else {              Toast.makeText(AddMovieActivity.this,                      "One or more fields left empty!",                      Toast.LENGTH\_LONG).show();            }          }        /\*\*       \* AsyncTask for adding a movie       \*/      private class AddMovieAsyncTask extends AsyncTask<String, String, String> {          @Override          protected void onPreExecute() {              super.onPreExecute();              //Display proggress bar              pDialog = new ProgressDialog(AddMovieActivity.this);              pDialog.setMessage("Adding Movie. Please wait...");              pDialog.setIndeterminate(false);              pDialog.setCancelable(false);              pDialog.show();          }            @Override          protected String doInBackground(String... params) {              HttpJsonParser httpJsonParser = new HttpJsonParser();              Map<String, String> httpParams = new HashMap<>();              //Populating request parameters              httpParams.put(KEY\_MOVIE\_NAME, movieName);              httpParams.put(KEY\_GENRE, genre);              httpParams.put(KEY\_YEAR, year);              httpParams.put(KEY\_RATING, rating);              JSONObject jsonObject = httpJsonParser.makeHttpRequest(                      BASE\_URL + "add\_movie.php", "POST", httpParams);              try {                  success = jsonObject.getInt(KEY\_SUCCESS);              } catch (JSONException e) {                  e.printStackTrace();              }              return null;          }            protected void onPostExecute(String result) {              pDialog.dismiss();              runOnUiThread(new Runnable() {                  public void run() {                      if (success == 1) {                          //Display success message                          Toast.makeText(AddMovieActivity.this,                                  "Movie Added", Toast.LENGTH\_LONG).show();                          Intent i = getIntent();                          //send result code 20 to notify about movie update                          setResult(20, i);                          //Finish ths activity and go back to listing activity                          finish();                        } else {                          Toast.makeText(AddMovieActivity.this,                                  "Some error occurred while adding movie",                                  Toast.LENGTH\_LONG).show();                        }                  }              });          }      }  } |

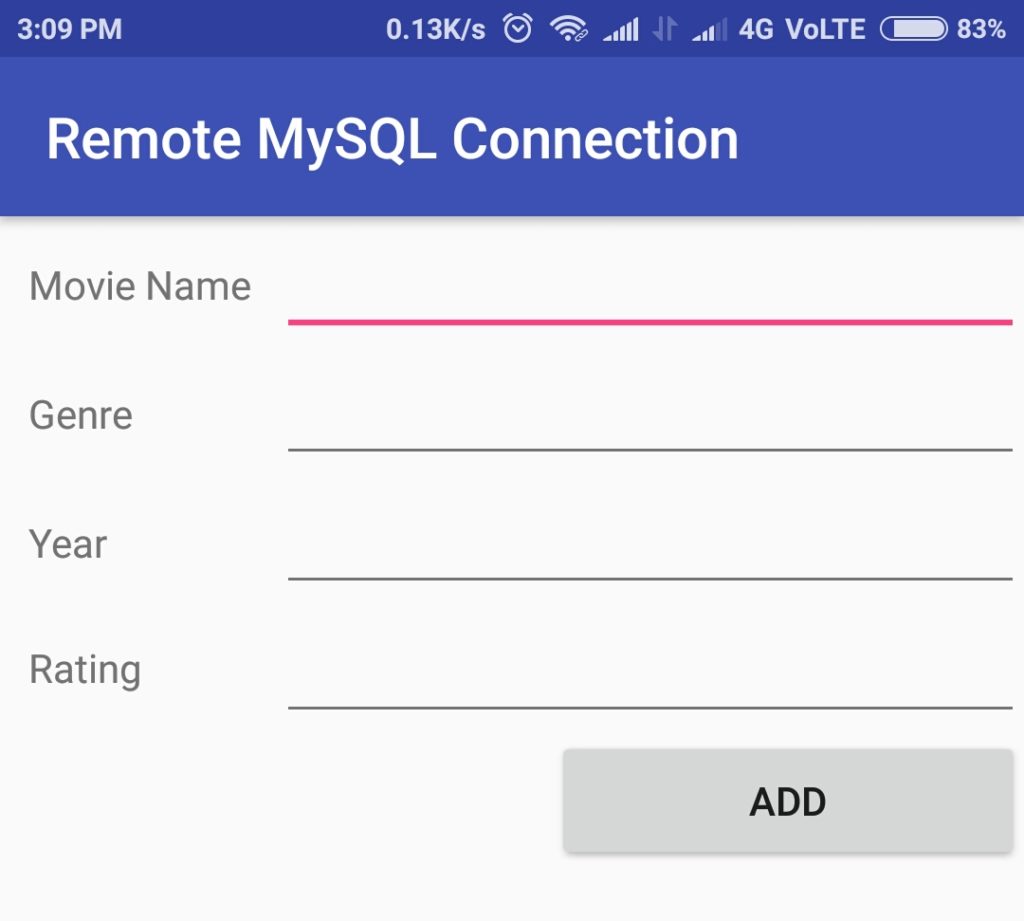
This activity helps in adding a new movie to the remote database.

activity\_add\_movie.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116 | <?xml version="1.0" encoding="utf-8"?>  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="wrap\_content">        <LinearLayout          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:orientation="vertical">            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.25"                  android:text="@string/movieName" />                <EditText                  android:layout\_width="0dp"                  android:id="@+id/txtMovieNameAdd"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10" />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.25"                  android:text="@string/genre" />                <EditText                  android:layout\_width="0dp"                  android:id="@+id/txtGenreAdd"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10" />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.25"                  android:text="@string/year" />                <EditText                  android:id="@+id/txtYearAdd"                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10" />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal"              android:paddingLeft="10sp">                <TextView                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_gravity="center\_vertical"                  android:layout\_weight="0.25"                    android:text="@string/rating" />                  <EditText                  android:id="@+id/txtRatingAdd"                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.75"                  android:ems="10"                    />          </LinearLayout>            <LinearLayout              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:orientation="horizontal">                  <View                  android:layout\_width="0dp"                  android:layout\_height="match\_parent"                  android:layout\_weight="0.50" />                <Button                  android:id="@+id/btnAdd"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.20"                  android:text="@string/add" />          </LinearLayout>        </LinearLayout>  </ScrollView> |



**12.** Finally, update **RemoteMySQLActivity**with the following code. It calls **MovieListingActivity** and **AddMovieActivity**whenever the respective button is clicked.

RemoteMySQLActivity.java

Java



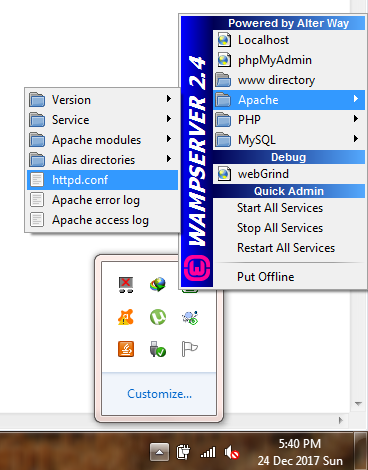
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | package com.androiddeft.remotemysqlconnection;    import android.content.Intent;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;  import android.widget.Toast;    import com.androiddeft.remotemysqlconnection.helper.CheckNetworkStatus;    public class RemoteMySQLActivity extends AppCompatActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_remote\_my\_sql);          Button viewAllBtn = (Button) findViewById(R.id.viewAllBtn);          Button addNewBtn = (Button) findViewById(R.id.addNewBtn);          viewAllBtn.setOnClickListener(new View.OnClickListener() {              @Override              public void onClick(View view) {                  //Check for network connectivity                  if (CheckNetworkStatus.isNetworkAvailable(getApplicationContext())) {                      Intent i = new Intent(getApplicationContext(),                              MovieListingActivity.class);                      startActivity(i);                  } else {                      //Display error message if not connected to internet                      Toast.makeText(RemoteMySQLActivity.this,                              "Unable to connect to internet",                              Toast.LENGTH\_LONG).show();                    }                }          });            addNewBtn.setOnClickListener(new View.OnClickListener() {              @Override              public void onClick(View view) {                  //Check for network connectivity                  if (CheckNetworkStatus.isNetworkAvailable(getApplicationContext())) {                      Intent i = new Intent(getApplicationContext(),                              AddMovieActivity.class);                      startActivity(i);                  } else {                      //Display error message if not connected to internet                      Toast.makeText(RemoteMySQLActivity.this,                              "Unable to connect to internet",                              Toast.LENGTH\_LONG).show();                    }                }          });        }    } |

**Testing on Android Device**

In this tutorial, I have used **192.168.43.72**as the base URL, which is my local IP address. You can replace this with your IP address or remote URL.

In order to test your application on your Android device,

**1.** Open **httpd.conf** file of your WAMP Installation (Located in D:\wamp\bin\apache\Apache2.4.4\conf in my case). You can also open it from the Notification Area Icons:



Now Add the highlighted line inside **Directory**tag:



|  |  |
| --- | --- |
| 1  2  3  4 | <Directory "D:/wamp/www">       Order Deny,Allow       Allow from all  </Directory> |

This allows the WAMP server to be accessed throughout the network.

**2.** Turn on WiFi Hotspot on your device and connect your device to PC.  Now you should be able to test the application.