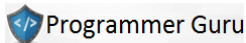


You know you can create Android Color
XMLs effortlessly



[Home](#) [Basics »](#) [Write For Us](#) [Advertise With Us](#) [Demo Videos](#) [Other Tutorials »](#) [About Us](#) [Contact Us](#)

66
Shares



Home

Web Service

Android Restful Webservice Tutorial – How to call RESTful webservice in Android – Part 3

Android Restful Webservice Tutorial – How to call RESTful webservice in Android – Part 3

Posted By Android Guru on May 27, 2014 | 32 Comments

In the Android RESTful tutorial series , I am discussing about creating and invoking **RESTful webservice** in Android applications.

In this particular post, I am going to discuss about how to invoke RESTful webservice (that has been created and deployed in local application server) from inside Android applications.

I already discussed about What is RESTful webservice and how to create it in couple of posts earlier. Have listed those posts links below, it would be helpful for you if you quickly take a look at those articles.

If you are new to RESTful webservice, I would recommend you to take a look at [Introduction to RESTful webservice](#)

This tutorial series will be published as three parts:

- Part 1 – [Introduction to RESTful webservice](#)
- Part 2 – [How to create RESTful webservice in Java?](#)
- Part 3 – [How to invoke RESTful webservice in Android applications?](#)

In this post, I will be discussing about invoking **RESTful webservice** we already created in the previous post (Read: [How to create RESTful webservice in Java?](#)).

If you haven't created the RESTful webservice yet, please go ahead and create it before proceeding with below steps.

Video Demo

[Android RESTful Webservice Demo](#)



Steps involved in invoking RESTful webservice in Android application:

By looking at the video demo, you got to know what are we going to develop.

About application

- It is a simple application to register and login the users.
- During registration and logging-in, user credentials (Username and Password) are taken to Server using RESTful webservice to register or authenticate the User.

You can download source code from [here](#) if you don't want to create Application from scratch, otherwise please proceed with below listings.

66
Shares**Step 1: Create Android Application Project**

Create new android project [File >> New >> Android Application Project] with project name

AndroidRestFulWSExample

Enter package name as 'com.prgguru.example'

Choose Minimum Required SDK, Target SDK and Compile with. Confused on choosing these options? Take a look at [Minimum Required SDK – Target SDK – Compile With](#) post.

Click Next button and finally click 'Finish' to create project

Step 2: Add library to project

Add below third party library into project's 'lib' folder. You can also download it if you don't have it with you.

[Android Asynchronous Http Client](#) – An asynchronous callback-based Http client for Android built on top of Apache's HttpClient libraries which is used by Pinterest, Instagram etc... [Download](#)

Step 3: Design Screens

We are going to have three activities:

HomeActivity – Home Screen of the Application. Launched once after successful login.

LoginActivity – Login screen of the application

RegisterActivity – Register screen of the application

Before creating screens, make sure below String resources are added. Add string resources to strings.xml present under /res/values folder.

[strings.xml](#)

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3
4      <string name="app_name">AndroidRestFulWSExample</string>
5      <string name="hello_world">Hello world!</string>
6      <string name="action_settings">Settings</string>
7      <string name="title_activity_login">Login</string>
8      <string name="title_activity_home">HomeActivity</string>
9      <string name="register_title">Register</string>
10     <string name="login_title">Login</string>
11     <string name="name">Name</string>
12     <string name="email">Email</string>
13     <string name="pwd">Password</string>
14     <string name="btnRegister">Register</string>
15 </resources>

```

Create three layout XMLs under /res/layout folder:

[home.xml](#)

Home Screen of the application that will be launched once after successful login by User.

```

1  <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:id="@+id/container"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context="com.prgguru.example.HomeActivity"
7      tools:ignore="MergeRootFrame" >
8
9      <TextView
10         android:layout_width="fill_parent"
11         android:layout_height="fill_parent"
12         android:layout_gravity="fill_vertical"
13         android:layout_marginTop="20pt"
14         android:gravity="center_horizontal"
15         android:text="Welcome User"
16         android:textSize="25dip" />
17
18 </FrameLayout>

```

[login.xml](#)

Login screen of the application.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="fill_parent"
4      android:layout_height="fill_parent" >

```

```

5
6
7 <LinearLayout
8     android:layout_width="fill_parent"
9     android:layout_height="wrap_content"
10    android:orientation="vertical"
11    android:padding="10dip" >
12
13    <TextView
14        android:layout_width="fill_parent"
15        android:layout_height="wrap_content"
16        android:layout_marginBottom="10dip"
17        android:text="@string/login_title"
18        android:textSize="25sp"
19        android:textStyle="bold" />
20
21    <EditText
22        android:id="@+id/loginEmail"
23        android:layout_width="fill_parent"
24        android:layout_height="wrap_content"
25        android:hint="Enter your Email ID"
26        android:inputType="textEmailAddress" />
27
28    <TextView
29        android:layout_width="fill_parent"
30        android:layout_height="wrap_content"
31        android:layout_marginTop="15dip"
32        android:text="@string/pwd" />
33
34    <EditText
35        android:id="@+id/loginPassword"
36        android:layout_width="fill_parent"
37        android:layout_height="wrap_content"
38        android:hint="Enter Password"
39        android:inputType="textPassword" />
40
41    <TextView
42        android:id="@+id/login_error"
43        android:layout_width="fill_parent"
44        android:layout_height="wrap_content"
45        android:padding="10dip"
46        android:textColor="#e30000"
47        android:textStyle="bold" />
48
49    <Button
50        android:id="@+id/btnLogin"
51        android:layout_width="fill_parent"
52        android:layout_height="wrap_content"
53        android:layout_marginTop="20dip"
54        android:background="#ff6347"
55        android:onClick="loginUser"
56        android:text="Login"
57        android:textColor="#fff" />
58
59    <Button
60        android:id="@+id/btnLinkToRegisterScreen"
61        android:layout_width="fill_parent"
62        android:layout_height="wrap_content"
63        android:layout_marginTop="40dip"
64        android:background="@null"
65        android:onClick="navigatetoRegisterActivity"
66        android:text="Signup"
67        android:textColor="#228b22"
68        android:textStyle="bold" />
69 </LinearLayout>
70 </ScrollView>

```

register.xml

Registration screen of the application.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent" >
5
6     <LinearLayout
7         android:layout_width="fill_parent"
8         android:layout_height="wrap_content"
9         android:orientation="vertical"
10        android:padding="5dip" >
11
12        <!-- View Title Label -->
13
14        <TextView
15            android:layout_width="fill_parent"
16            android:layout_height="wrap_content"
17            android:layout_marginBottom="10dip"
18            android:text="@string/register_title"
19            android:textSize="25sp"
20            android:textStyle="bold" />
21        <!-- Name Label -->
22
23        <TextView
24            android:layout_width="fill_parent"
25            android:layout_height="wrap_content"
26            android:text="@string/name" />
27        <!-- Name TextField -->
28
29        <EditText
30            android:id="@+id/registerName"
31            android:layout_width="fill_parent"
32            android:layout_height="wrap_content" android:hint="Enter your Name"/
33
34        <!-- Email Label -->
35

```

66
Shares

```

36         <TextView
37             android:layout_width="fill_parent"
38             android:layout_height="wrap_content"
39             android:text="@string/email" />
40         <!-- Email TextField -->
41
42         <EditText
43             android:id="@+id/registerEmail"
44             android:layout_width="fill_parent"
45             android:layout_height="wrap_content"
46             android:hint="Enter your Email ID"
47             android:inputType="textEmailAddress" />
48
49         <!-- Password Label -->
50
51         <TextView
52             android:layout_width="fill_parent"
53             android:layout_height="wrap_content"
54             android:layout_marginTop="15dip"
55             android:text="@string/pwd" />
56         <!-- Password TextField -->
57
58         <EditText
59             android:id="@+id/registerPassword"
60             android:layout_width="fill_parent"
61             android:layout_height="wrap_content"
62             android:hint="Enter Password"
63             android:inputType="textPassword" />
64
65         <!-- Error message -->
66
67         <TextView
68             android:id="@+id/register_error"
69             android:layout_width="fill_parent"
70             android:layout_height="wrap_content"
71             android:gravity="center"
72             android:padding="10dip"
73             android:textColor="#e30000"
74             android:textStyle="bold" />
75
76         <Button
77             android:id="@+id/btnRegister"
78             android:layout_width="fill_parent"
79             android:layout_height="wrap_content"
80             android:layout_marginTop="20dip"
81             android:background="#ff6347"
82             android:onClick="registerUser"
83             android:text="@string/btnRegister"
84             android:textColor="#fff" />
85
86         <!-- Link to Login Screen -->
87
88         <Button
89             android:id="@+id/btnLinkToLoginScreen"
90             android:layout_width="fill_parent"
91             android:layout_height="wrap_content"
92             android:layout_marginTop="20dip"
93             android:background="@null"
94             android:onClick="navigatetoLoginActivity"
95             android:text="Login"
96             android:textColor="#228b22"
97             android:textStyle="bold" />
98     </LinearLayout>
99 </ScrollView>
100

```

Step 4: Utility.java – Utility Class

We are done with Layout designing, let us jump into Coding.

Create Utility.java under the package com.prgguru.example and fill it with below code. It has Utility methods to be used in other Activity classes.

```

1  package com.prgguru.example;
2
3  import java.util.regex.Matcher;
4  import java.util.regex.Pattern;
5  /**
6   * Class which has Utility methods
7   *
8   */
9  public class Utility {
10     private static Pattern pattern;
11     private static Matcher matcher;
12     //Email Pattern
13     private static final String EMAIL_PATTERN =
14         "[A-Za-z0-9-\\+](\\.[A-Za-z0-9-\\+]*@|
15         + "[A-Za-z0-9-\\+](\\.[A-Za-z0-9-\\+]*\\.[A-Za-z]{2,})$";
16
17     /**
18      * Validate Email with regular expression
19      *
20      * @param email
21      * @return true for Valid Email and false for Invalid Email
22      */
23     public static boolean validate(String email) {
24         pattern = Pattern.compile(EMAIL_PATTERN);
25         matcher = pattern.matcher(email);
26         return matcher.matches();
27     }
28
29     /**
30      * Checks for Null String object
31      *
32      * @param txt

```

```

33     * @return true for not null and false for null String object
34     */
35     public static boolean isNotNull(String txt){
36         return txt!=null && txt.trim().length()>0 ? true: false;
37     }
38 }

```

Step 5: Home.java – Home Activity Class

Home screen Activity which will be launched once after successful login by User.

```

1 package com.prgguru.example;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 /**
6  *
7  * Home Screen Activity
8  */
9 public class HomeActivity extends Activity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         //Displays Home Screen
15         setContentView(R.layout.home);
16     }
17
18 }

```

Step 6: LoginActivity.java – Login Screen Activity Class

Login screen which will allow User to login to Application. Make sure the IP address has been changed in the AsyncHttp Get method with localhost or IP address of your machine.

LoginActivity.java is well commented, so by reading the comments you will understand the meaning of it. In case, if you still have question [32 Comments](#) it right away.

```

1 package com.prgguru.example;
2
3 import org.json.JSONException;
4 import org.json.JSONObject;
5
6 import android.app.Activity;
7 import android.app.ProgressDialog;
8 import android.content.Intent;
9 import android.os.Bundle;
10 import android.view.View;
11 import android.widget.EditText;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import com.loopj.android.http.AsyncHttpClient;
16 import com.loopj.android.http.AsyncHttpResponseHandler;
17 import com.loopj.android.http.RequestParams;
18 /**
19  *
20  * Login Activity Class
21  */
22
23 public class LoginActivity extends Activity {
24     // Progress Dialog Object
25     ProgressDialog prgDialog;
26     // Error Msg TextView Object
27     TextView errorMsg;
28     // Email Edit View Object
29     EditText emailET;
30     // Passwprd Edit View Object
31     EditText pwdET;
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.login);
36         // Find Error Msg Text View control by ID
37         errorMsg = (TextView)findViewById(R.id.login_error);
38         // Find Email Edit View control by ID
39         emailET = (EditText)findViewById(R.id.loginEmail);
40         // Find Password Edit View control by ID
41         pwdET = (EditText)findViewById(R.id.loginPassword);
42         // Instantiate Progress Dialog object
43         prgDialog = new ProgressDialog(this);
44         // Set Progress Dialog Text
45         prgDialog.setMessage("Please wait...");
46         // Set Cancelable as False
47         prgDialog.setCancelable(false);
48     }
49
50     /**
51     * Method gets triggered when Login button is clicked
52     *
53     * @param view
54     */
55     public void loginUser(View view){
56         // Get Email Edit View Value
57         String email = emailET.getText().toString();
58         // Get Password Edit View Value
59         String password = pwdET.getText().toString();
60         // Instantiate Http Request Param Object
61         RequestParams params = new RequestParams();
62         // When Email Edit View and Password Edit View have values other than Nu
63         if(Utility.isNotNull(email) && Utility.isNotNull(password)){
64             // When Email entered is Valid
65             if(Utility.validate(email)){
66                 // Put Http parameter username with value of Email Edit View cor

```

```

67         params.put("username", email);
68         // Put Http parameter password with value of Password Edit Value
69         params.put("password", password);
70         // Invoke RESTful Web Service with Http parameters
71         invokeWS(params);
72     }
73     // When Email is invalid
74     else{
75         Toast.makeText(getApplicationContext(), "Please enter valid email",
76             Toast.LENGTH_SHORT).show();
77     }
78     else{
79         Toast.makeText(getApplicationContext(), "Please fill the form, don't
80         leave any field empty", Toast.LENGTH_SHORT).show();
81     }
82 }
83 /**
84  * Method that performs RESTful webservice invocations
85  *
86  * @param params
87  */
88 public void invokeWS(RequestParams params){
89     // Show Progress Dialog
90     prgDialog.show();
91     // Make RESTful webservice call using AsyncHttpClient object
92     AsyncHttpClient client = new AsyncHttpClient();
93     client.get("http://192.168.2.2:9999/useraccount/login/dologin", params,
94         // When the response returned by REST has Http response code '200'
95         @Override
96         public void onSuccess(String response) {
97             // Hide Progress Dialog
98             prgDialog.hide();
99             try {
100                 // JSON Object
101                 JSONObject obj = new JSONObject(response);
102                 // When the JSON response has status boolean value assigned
103                 if(obj.getBoolean("status")){
104                     Toast.makeText(getApplicationContext(), "You are successfully
105                     logged in", Toast.LENGTH_SHORT).show();
106                     // Navigate to Home screen
107                     navigatetoHomeActivity();
108                 }
109                 // Else display error message
110                 else{
111                     errorMsg.setText(obj.getString("error_msg"));
112                     Toast.makeText(getApplicationContext(), obj.getString("error_msg"),
113                         Toast.LENGTH_SHORT).show();
114                 }
115             } catch (JSONException e) {
116                 // TODO Auto-generated catch block
117                 Toast.makeText(getApplicationContext(), "Error Occured [Server side]",
118                     Toast.LENGTH_SHORT).show();
119             }
120         }
121     // When the response returned by REST has Http response code other than 200
122     @Override
123     public void onFailure(int statusCode, Throwable error,
124         String content) {
125         // Hide Progress Dialog
126         prgDialog.hide();
127         // When Http response code is '404'
128         if(statusCode == 404){
129             Toast.makeText(getApplicationContext(), "Requested resource not found",
130                 Toast.LENGTH_SHORT).show();
131         }
132         // When Http response code is '500'
133         else if(statusCode == 500){
134             Toast.makeText(getApplicationContext(), "Something went wrong",
135                 Toast.LENGTH_SHORT).show();
136         }
137         // When Http response code other than 404, 500
138         else{
139             Toast.makeText(getApplicationContext(), "Unexpected Error Occured",
140                 Toast.LENGTH_SHORT).show();
141         }
142     }
143 }
144 /**
145  * Method which navigates from Login Activity to Home Activity
146  */
147 public void navigatetoHomeActivity(){
148     Intent homeIntent = new Intent(getApplicationContext(), HomeActivity.class);
149     homeIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
150     startActivity(homeIntent);
151 }
152 /**
153  * Method gets triggered when Register button is clicked
154  *
155  * @param view
156  */
157 public void navigatetoRegisterActivity(View view){
158     Intent loginIntent = new Intent(getApplicationContext(), RegisterActivity.class);
159     loginIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
160     startActivity(loginIntent);
161 }
162 }

```

Step 7: RegisterActivity.java – RegisterScreen Activity Class

Register screen which will allow User to register him/herself to the Application. Make sure the IP address has been changed in the AsyncHttp Get method with localhost or IP address of your machine:

RegisterActivity.java is well commented, so by reading the comments you will understand the meaning of it. In case, if you still have question [32 Comments](#) it right away.

66
Shares

```

1 package com.prgguru.example;
2
3 import org.json.JSONException;
4 import org.json.JSONObject;
5
6 import android.app.Activity;
7 import android.app.ProgressDialog;
8 import android.content.Intent;
9 import android.os.Bundle;
10 import android.view.View;
11 import android.widget.EditText;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import com.loopj.android.http.AsyncHttpClient;
16 import com.loopj.android.http.AsyncHttpResponseHandler;
17 import com.loopj.android.http.RequestParams;
18 /**
19  *
20  * Register Activity Class
21  */
22 public class RegisterActivity extends Activity {
23     // Progress Dialog Object
24     ProgressDialog prgDialog;
25     // Error Msg TextView Object
26     TextView errorMsg;
27     // Name Edit View Object
28     EditText nameET;
29     // Email Edit View Object
30     EditText emailET;
31     // Passwprd Edit View Object
32     EditText pwdET;
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36         setContentView(R.layout.register);
37         // Find Error Msg Text View control by ID
38         errorMsg = (TextView)findViewById(R.id.register_error);
39         // Find Name Edit View control by ID
40         nameET = (EditText)findViewById(R.id.registerName);
41         // Find Email Edit View control by ID
42         emailET = (EditText)findViewById(R.id.registerEmail);
43         // Find Password Edit View control by ID
44         pwdET = (EditText)findViewById(R.id.registerPassword);
45         // Instantiate Progress Dialog object
46         prgDialog = new ProgressDialog(this);
47         // Set Progress Dialog Text
48         prgDialog.setMessage("Please wait...");
49         // Set Cancelable as False
50         prgDialog.setCancelable(false);
51     }
52
53     /**
54      * Method gets triggered when Register button is clicked
55      *
56      * @param view
57      */
58     public void registerUser(View view){
59         // Get NAME ET control value
60         String name = nameET.getText().toString();
61         // Get Email ET control value
62         String email = emailET.getText().toString();
63         // Get Password ET control value
64         String password = pwdET.getText().toString();
65         // Instantiate Http Request Param Object
66         RequestParams params = new RequestParams();
67         // When Name Edit View, Email Edit View and Password Edit View have val
68         if(Utility.isNotNull(name) && Utility.isNotNull(email) && Utility.isNotN
69             // When Email entered is Valid
70             if(Utility.validate(email)){
71                 // Put Http parameter name with value of Name Edit View control
72                 params.put("name", name);
73                 // Put Http parameter username with value of Email Edit View cor
74                 params.put("username", email);
75                 // Put Http parameter password with value of Password Edit View
76                 params.put("password", password);
77                 // Invoke RESTful Web Service with Http parameters
78                 invokeWS(params);
79             }
80             // When Email is invalid
81             else{
82                 Toast.makeText(getApplicationContext(), "Please enter valid emai
83             }
84         }
85         // When any of the Edit View control left blank
86         else{
87             Toast.makeText(getApplicationContext(), "Please fill the form, don't
88         }
89     }
90 }
91
92 /**
93  * Method that performs RESTful webservice invocations
94  *
95  * @param params
96  */
97     public void invokeWS(RequestParams params){
98         // Show Progress Dialog
99         prgDialog.show();
100         // Make RESTful webservice call using AsyncHttpClient object
101         AsyncHttpClient client = new AsyncHttpClient();
102         client.get("http://192.168.2.2:9999/useraccount/register/doregister", par
103             // When the response returned by REST has Http response code '200'
104             @Override
105             public void onSuccess(String response) {
106                 // Hide Progress Dialog
107                 prgDialog.hide();

```

66
Shares

```

108         try {
109             // JSON Object
110             JSONObject obj = new JSONObject(response);
111             // When the JSON response has status boolean value assi
112             if(obj.getBoolean("status")){
113                 // Set Default Values for Edit View controls
114                 setDefaultValues();
115                 // Display successfully registered message using To
116                 Toast.makeText(getApplicationContext(), "You are su
117             }
118             // Else display error message
119             else{
120                 errorMsg.setText(obj.getString("error_msg"));
121                 Toast.makeText(getApplicationContext(), obj.getStri
122             }
123         } catch (JSONException e) {
124             // TODO Auto-generated catch block
125             Toast.makeText(getApplicationContext(), "Error Occured [Ser
126             e.printStackTrace();
127         }
128     }
129     // When the response returned by REST has Http response code other
130     @Override
131     public void onFailure(int statusCode, Throwable error,
132         String content) {
133         // Hide Progress Dialog
134         prgDialog.hide();
135         // When Http response code is '404'
136         if(statusCode == 404){
137             Toast.makeText(getApplicationContext(), "Requested resource
138         }
139         // When Http response code is '500'
140         else if(statusCode == 500){
141             Toast.makeText(getApplicationContext(), "Something went wr
142         }
143         // When Http response code other than 404, 500
144         else{
145             Toast.makeText(getApplicationContext(), "Unexpected Error c
146         }
147     }
148 }
149 }
150 }
151 }
152 /**
153  * Method which navigates from Register Activity to Login Activity
154  */
155 public void navigatetoLoginActivity(View view){
156     Intent loginIntent = new Intent(getApplicationContext(), LoginActivity.cl
157     // Clears History of Activity
158     loginIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
159     startActivity(loginIntent);
160 }
161 /**
162  * Set default values for Edit View controls
163  */
164 public void setDefaultValues(){
165     nameET.setText("");
166     emailET.setText("");
167     pwdET.setText("");
168 }
169 }
170 }
171 }

```

Step 8: Add Activities in AndroidManifest.xml

Make sure you added activities we created in AndroidManifest.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.prgguru.example"
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7      <uses-sdk
8          android:minSdkVersion="8"
9          android:targetSdkVersion="19" />
10
11     <uses-permission android:name="android.permission.INTERNET" />
12
13     <application
14         android:allowBackup="true"
15         android:icon="@drawable/ic_launcher"
16         android:label="@string/app_name"
17         android:theme="@style/AppTheme" >
18         <activity
19             android:name="com.prgguru.example.RegisterActivity"
20             android:label="@string/app_name" >
21             <intent-filter>
22                 <action android:name="android.intent.action.MAIN" />
23                 <category android:name="android.intent.category.LAUNCHER" />
24             </intent-filter>
25         </activity>
26         <activity
27             android:name="com.prgguru.example.LoginActivity"
28             android:label="@string/title_activity_login" >
29         </activity>
30         <activity
31             android:name="com.prgguru.example.HomeActivity"
32             android:label="@string/title_activity_home" >
33         </activity>
34     </application>
35 </manifest>
36
37

```


Step 9: Add Permission in AndroidManifest.xml

Don't forget to add internet permission in AndroidManifest.xml:

```
1 | <!-- Permission: Allow application to connect to Internet -->
2 | <uses-permission android:name="android.permission.INTERNET" />
```

Error Handling

When the Http response code returned by REST web service is other 200, which means an error might have occurred at server end. Error can be easily handled inside onFailure() method of AsyncHttpClient's post method as shown below:

66 Shares

```
1 | @Override
2 | public void onFailure(int statusCode, Throwable error,
3 | String content) {
4 |     // Hide Progress Dialog
5 |     prgDialog.hide();
6 |     // When Http response code is '404'
7 |     if(statusCode == 404){
8 |         Toast.makeText(getApplicationContext(), "Requested resource not four
9 |     }
10 |    // When Http response code is '500'
11 |    else if(statusCode == 500){
12 |        Toast.makeText(getApplicationContext(), "Something went wrong at ser
13 |    }
14 |    // When Http response code other than 404, 500
15 |    else{
16 |        Toast.makeText(getApplicationContext(), "Unexpected Error occurred!
17 |    }
18 | }
```

RESTful Webservice Response – JSON**Login**

When login is successful, the JSON returned by REST will look like:

```
1 | {
2 |   tag: "login"
3 |   status: true
4 | }
```

When login is unsuccessful (User entered incorrect Username or Password), the JSON returned by REST will look like:

```
1 | {
2 |   tag: "login"
3 |   status: false
4 |   error_msg: "Incorrect Email or Password"
5 | }
```

Registration

When registration is successful, the JSON returned by REST will look like:

```
1 | {
2 |   tag: "register"
3 |   status: true
4 | }
```

When registration is unsuccessful (User already registered), the JSON returned by REST will look like:

```
1 | {
2 |   tag: "register"
3 |   status: false
4 |   error_msg: "You are already registered"
5 | }
```

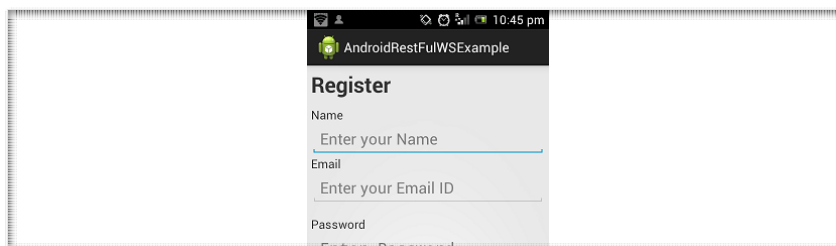
[pglinkadssmall1]

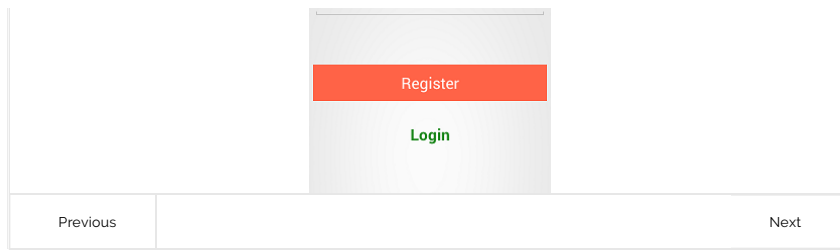
Demo

☐ Make sure the RESTful webservice you created is up and running.

That's all. It's time to test our code.

Run the application using emulator or device by right clicking on the project >> Run as >> Android applicaiton >> Choose emulator or device.





66

Shares



Download Source Code

Entire project is zipped and is available for download. Unzip the downloaded project and to import the project into eclipse, launch eclipse >> File >> Import.. >> Choose downloaded project([How to import android project in eclipse](#)).

[Download Source Code](#)

*apk in Android is the installation file similar to exe in windows.

[pglinkadssmall]

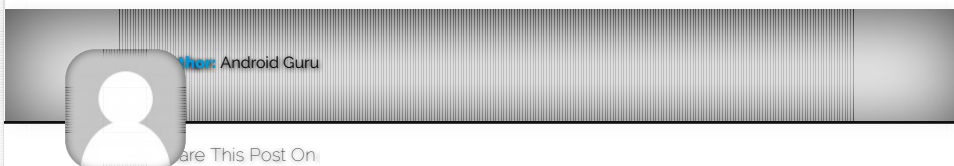
If you feel this article is helpful and interesting please spread a word about it to your friends and colleagues by sharing the article in Facebook or Twitter.

Tweet **Share** 44

You are always welcome to provide your comments and feedback from [32 Comments](#) box.

[pgwriteforus]

[pgfeedback]



32 Comments Programmer Guru Tutorial Blog Login

Recommend 1 Share Sort by Newest

Join the discussion...

Kavya • 6 months ago
Excellent post ! A good beginning point to start off.

I created java app described in part 2 and could successfully run and add user to mySQL db.I created the android app,but i am getting the error when i try to register a user on android emulator :
Unexpected Error occurred! [Most common Error: Device might not be connected....
I checked the statuscode, it is zero.The service is up and running too(In eclipse I clicked on java web project and clicked "Run as" Run on server) I cannot figure out the problem.Can you help me out..
Thanks
66 Shares
f t g+
• Reply • Share

Hnib • 6 months ago
Hi Guru,

It is a best tut for Restful Webservice. I can run the app normally on my device with the host is my computer. I have a host free, I do not know how I make the web service restful on this host. May you help me ? Thanks.
• Reply • Share

Prasanth • a year ago
Hi Thanks for the tutorial and i have tried this but ended up with an error and i am unable to solve the error

Authentication scheme ntlm not supported.
Unable to respond to any of these challenges: {ntlm=Proxy-Authenticate: Kerberos, ntlm=Proxy-Authenticate: NTLM, negotiate=Proxy-Authenticate: Negotiate}

Note : i am having proxy.

Could you please help me in solving this issue..?
• Reply • Share

Juan • a year ago
Amazing tutorial. I did it instead i'm newbie of Java & Android.

At this time i want to create this Android app which can do
1. A web-service supply XML Result from a query (select some items in my list. Ex: select no_cou, name_cou from countries).
2. A Android, execute above web-service, this list in own-screen.

Do you any suggestion?

Thank you very much.
• Reply • Share

Hector • a year ago
Hi, first thanks for the really good tutorial, I just have one problem when I run the app on the android emulator: when i click on register the apps exits with an error and when i look at the log i see this error "12-11 17:46:33.444: E/AndroidRuntime(2752): Suppressed: java.lang.ClassNotFoundException: com.loopj.android.http.RequestParams"

i know the import is done correctly because i don't get any error when compiling the code, is at run time that i get this error, do you know what might be happening?
• Reply • Share

Android Guru Mod → Hector • a year ago
Make sure loopj library is inside libs folder and also you are packing it inside apk.
• Reply • Share

Sudhir Kumar • a year ago
hey sir.....this error "Unexpected Error occurred! [Most common Error: Device might not be connected to Internet or remote server is not up and running]", given the service.....
what's reason...
• Reply • Share

Dinuka • a year ago
Hello sir, is it possible to connect this to a Google App Engine? A Google cloud instance? This is by far the best tutorial I've come across so if I connect this to my Google Storage, it would be epic! :D I do have an appspot domain as well but the app returns the following message "Server's JSON response might be invalid"
1 • Reply • Share

Android Guru Mod → Dinuka • a year ago
It must work. Make sure you construct valid JSON in Appspot server,
• Reply • Share

Sir Jones • a year ago
Hey my Friend really a nice nice Tutorial. and a nice Website to. Can you tell me why you use the Http Client from James Smith instead the "normal" HttpURLConnection? THX
• Reply • Share

Android Guru Mod → Sir Jones • a year ago
Am using Asynchronous Http Client - An asynchronous callback-based Http client for Android built on top of Apache's HttpClient libraries which is used by Pinterest, Instagram etc.,
• Reply • Share

RaMESH • a year ago
I am always getting Unexpected Error occurred! [Most common Error: Device might not be connected to Internet or remote server is not up and running]

Search this website

powered by Google

Categories

66 Shares Categories

Select Category

Follow Us

43.1k Follows

f Facebook 18.1k Followers	g+ Google+ 1.2k Followers
t Twitter 339 Followers	p Pinterest 44 Followers
RSS 23k Followers	YouTube 344 Followers

Recent Popular Random

SnackBar Vs Toast

Feb 26, 2016

Introduction to Android Fragments

Feb 25, 2016

6 Most Fascinating Emulators for Android Devices

Jun 5, 2015

Android GUI Tools and Kits you can't afford to miss as an app developer

Jan 15, 2015

Tags

Action Bar Activity AJAX android apps development **Android Basics** android development company androidversions Android Wear Authentication Broadcast receivers Button Call Camera Dot Net Web Service Eclipse Email Facebook Gallery GCM Image Image Upload Intent Java Web Service jquery jQuery Mobile JSON Layout Library Localization Media Notifications php Restful webservice SD Card Seek bar SQLite Text to speech Vibrate Video web Web service Webview

Recent Posts

- ▶ SnackBar Vs Toast February 26, 2016
- ▶ Introduction to Android Fragments February 25, 2016
- ▶ 6 Most Fascinating Emulators for Android Devices June 5, 2015

Android Color Generator**About Me**

Welcome to ProgrammerGuruTutorial Blog. Let me introduce myself more formally. I'm a normal guy, engineer by education who is passionate about Programming and Internet. Here I am writing lot about jQuery, AJAX, Android and JSON. Want to join with me in the journey of learning aforementioned technology? You are in the right place then :) Keep me posted with your valuable feedback and comments. Happy learning

Applications AQuery Basics Facebook GCM JSON Layout Localization Media SQLite Toast Web Service

66

Shares

