

Slide 11.8: Java source code: SigninActivity.java (cont.)

Slide 11.10: Server-side PHP scripts

Home



## SigninActivity.java (Cont.)

---

Line 29: [protected String doInBackground\( String... arg0 \) {](#)

It invoked on the background thread immediately after [onPreExecute](#) finishes executing. This step is used to perform background computation that can take a long time. The specified parameters are the parameters passed to [execute\(Params...\)](#) by the caller of this task. The result of the computation must be returned by this step and will be passed back to the last step.

Line 38: [URLEncoder.encode\( username, "UTF-8" \);](#)

Translates a string into [application/x-www-form-urlencoded](#) format using the specific encoding scheme [UTF-8](#).

Line 47: [URLConnection conn = url.openConnection\( \);](#)

Returns a [URLConnection](#) instance that represents a connection to the remote object referred to by the URL.

Line 48: [conn.setDoOutput\( true\);](#)

Sets the value of the [doOutput](#) field for this [URLConnection](#) to the specified value. A URL connection can be used for input and/or output. Set the DoOutput flag to true if you intend to use the URL connection for output, false if not. The default is false.

Line 57: [conn.getOutputStream\( \) \);](#)

It returns an output stream that writes to this connection.

Line 59: [wr.flush\( \);](#)

Flushes the stream and forces any buffered output to be written out.

## Line 79: `protected void onPostExecute( String result ) {`

Runs on the UI thread after `doInBackground(Params...)`. The specified result is the value returned by `doInBackground(Params...)`.

Server/app/src/main/java/com/wenchen/server/SigninActivity.java

```

01 package com.wenchen.server;
02
03 import android.content.Context;
04 import android.os.AsyncTask;
05 import android.widget.TextView;
06 import java.net.URL;
07 import java.net.URLConnection;
08 import java.net.URLEncoder;
09 import java.io.BufferedReader;
10 import java.io.InputStreamReader;
11 import java.io.OutputStreamWriter;
12
13 public class SigninActivity extends AsyncTask<String, Void, String> {
14     private TextView statusField, roleField;
15     private Context context;
16     private int byGetOrPost = 0;
17
18     // Flag 0 means GET and 1 means POST. (By default it is GET.)
19     public SigninActivity( Context context, TextView statusField, TextView roleField,
20 int flag ) {
21         this.context = context;
22         this.statusField = statusField;
23         this.roleField = roleField;
24         byGetOrPost = flag;
25     }
26
27     protected void onPostExecute( ) { }
28
29     @Override
30     protected String doInBackground( String... arg0 ) {
31         try {
32             String username = (String) arg0[0];
33             String password = (String) arg0[1];
34             String link = "http://wenchen.cs.und.edu/course/457/11/Android/";
35
36             // Complete the URL.
37             if ( byGetOrPost == 0 ) { // Get method
38                 link += "login_get.php";
39                 link += "?username=" + URLEncoder.encode( username, "UTF-8" );
40                 link += "&password=" + URLEncoder.encode( password, "UTF-8" );
41             }
42             else { // Post method
43                 link += "login_post.php";
44             }
45
46             // Connect to the server.
47             URL url = new URL( link );
48             URLConnection conn = url.openConnection( );
49             conn.setDoOutput( true );
50
51             // Send the arguments via standard output for the POST method.
52             if ( byGetOrPost == 1 ) { // Post method
53                 String data = URLEncoder.encode( "username", "UTF-8" ) + "=";
54                 data += URLEncoder.encode( username, "UTF-8" ) + "&";
55             }
56         }
57     }
58 }

```

```
54         data += URLEncoder.encode( "password", "UTF-8" ) + "=";
55         data += URLEncoder.encode( password, "UTF-8" );
56         OutputStreamWriter wr = new OutputStreamWriter(
57             conn.getOutputStream( ) );
58         wr.write( data );
59         wr.flush( );
60     }
61
62     // Read server response.
63     BufferedReader reader = new BufferedReader(
64         new InputStreamReader( conn.getInputStream( ) ));
65     StringBuilder sb = new StringBuilder( );
66     String line = null;
67     while ( ( line = reader.readLine( ) ) != null ) {
68         sb.append( line );
69         break;
70     }
71     return sb.toString( );
72 }
73 catch( Exception e ) {
74     return new String( "Exception: " + e.getMessage( ) );
75 }
76 }
77
78 @Override
79 protected void onPostExecute( String result ) {
80     this.statusField.setText( "Login Successful" );
81     this.roleField.setText ( result );
82 }
83 }
```

Slide 11.8: Java source code: SigninActivity.java (cont.)

Slide 11.10: Server-side PHP scripts

[Home](#)

