

# COMP7506 Smart phone apps development

## Workshop 4 Android Apps Development (Part 3)

### 1 Introduction

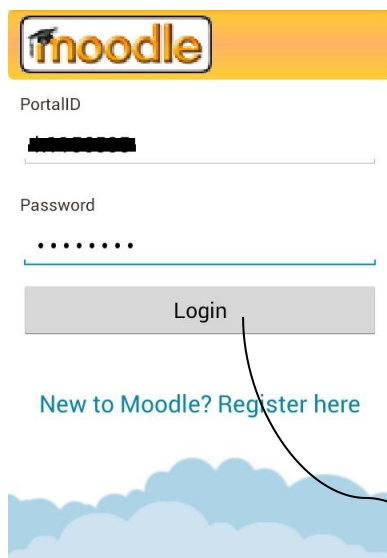
In workshop ONE and TWO, we have developed the basic framework of the app.

Now, we are going to get real data from **HKU Moodle** by using **HTTP** requests.

**AsyncTask** is used to setup the connection because android does not allow connection inside the main thread.

The data from Moodle is HTML source. In order to extract useful information (Course Title, Teachers), we are going to use **regular expression** to achieve the purpose.

Codes will be provided for reference!



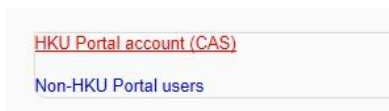
- 2 Importing the project of workshop 2
- 3 Login to hkuportal, and redirect to moodle

### 3.1 Introduction

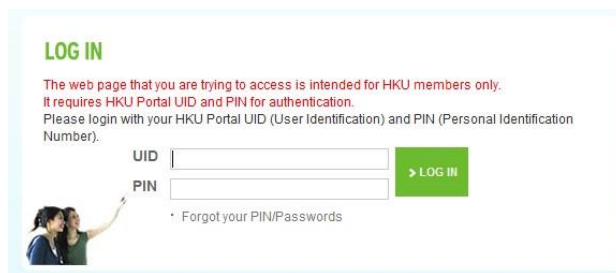
Handling HTTP request is easy with HttpClient, however this class has been deprecated since Android 5.0. Instead, URLConnection is used to handle HTTP and also HTTPS connections. In general, the most commonly used HTTP requests are POST and GET. In this workshop, we only need GET operation, which is the default calling method of URLConnection for HTTP. Students can refer to <http://developer.android.com/intl/ja/reference/java/net/URLConnection.html> for more details of URLConnection. In addition, a CookieManager has to be initialized in application to handle cookies when using URLConnection.

The flow chart of login moodle:

Step 1: <http://moodle.hku.hk>



Step 2: <https://hkuportal.hku.hk/cas/login?service=http%3A%2F%2Fmoodle.hku.hk%2Flogin%2Findex.php?authCAS||CAS>



The HTTP requests behind the browser

5	200	HTTPS	hkuportal.hku.hk	/cas/servlet/edu.yale.its.tp.cas.servlet.Login
6	302	HTTP	moodle.hku.hk	/login/index.php?authCAS=CAS&ticket= <b>[Ticket ID]</b>
7	303	HTTP	moodle.hku.hk	/login/index.php?authCAS=CAS
8	303	HTTP	moodle.hku.hk	/login/index.php?testsession=24535
9	200	HTTP	moodle.hku.hk	/

Ticket ID

First, we need to get a key id with keyed() function, it is the timestamp. Then we get a link as below form

```
String urlParameters = "keyid=" + keyed() +
"&service=https://moodle.hku.hk/login/index.php?authCAS=CAS&username=" + userName +
"&password=" + userPW + "&x=38&y=26";
```

The moodle authentication is simple. We just need to stick the ticket ID at the end of url.

```
"https://moodle.hku.hk/login/index.php?authCAS=CAS&ticket=" + ticketID
```

After authentication, moodle will return the HTML source of its first page.

### 3.2 In the MainActivity class, add the following three functions

```
public String ReadBufferedHTML(BufferedReader reader, char[] htmlBuffer, int bufSz) throws java.io.IOException {
    htmlBuffer[0] = '\0';
    int offset = 0;
    do {
        int cnt = reader.read(htmlBuffer, offset, bufSz - offset);
        if (cnt > 0) {
            offset += cnt;
        } else {
            break;
        }
    } while (true);
    return new String(htmlBuffer);
}
```

```
// generate keyid of POST data to hku portal
public String keyid() {
    Calendar c1 = Calendar.getInstance();
    String time = String.valueOf(c1.get(Calendar.YEAR)) + String.valueOf(c1.get(Calendar.MONTH))
        + String.valueOf(c1.get(Calendar.DATE)) + String.valueOf(c1.get(Calendar.HOUR))
        + String.valueOf(c1.get(Calendar.MINUTE)) + String.valueOf(c1.get(Calendar.SECOND));
    return time;
}

public String getMoodleFirstPage(String userName, String userPW) {
    HttpsURLConnection conn_portal = null;
    URLConnection conn_moodle = null;
    final int HTML_BUFFER_SIZE = 2 * 1024 * 1024;
    char htmlBuffer[] = new char[HTML_BUFFER_SIZE];
    final int HTTPCONNECTION_TYPE = 0;
    final int HTTPSCONNECTION_TYPE = 1;
    int moodle_conn_type = HTTPCONNECTION_TYPE;
    try {
        ////////////////////////////////// HKU portal //////////////////////////////////
        // URL url_portal = new
        // URL("https://hkuportal.hku.hk/cas/login?service=http://moodle.hku.hk/login/index.php?authCAS=CAS&username="
        // + userName + "&password=" + userPW);
        URL url_portal = new
            URL("https://hkuportal.hku.hk/cas/servlet/edu.yale.its.tp.cas.servlet.Login");
        conn_portal = (HttpsURLConnection) url_portal.openConnection();
        String urlParameters = "keyid=" + keyid() + "&service=https://moodle.hku.hk/login/index.php?authCAS=CAS&username="
            + userName + "&password=" + userPW + "&x=38&y=26";
        byte[] postData = urlParameters.getBytes(StandardCharsets.UTF_8);
        int postDataLength = postData.length;
        conn_portal.setDoOutput(true);
        conn_portal.setInstanceFollowRedirects(false);
        conn_portal.setRequestMethod("POST");
        conn_portal.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        conn_portal.setRequestProperty("charset", "utf-8");
        conn_portal.setRequestProperty("Content-Length", Integer.toString(postDataLength));
        conn_portal.setUseCaches(false);
        try (DataOutputStream wr = new DataOutputStream(conn_portal.getOutputStream())) {
            wr.write(postData);
        }
        BufferedReader reader_portal = new BufferedReader(new InputStreamReader(conn_portal.getInputStream()));
        String HTMLSource = ReadBufferedHTML(reader_portal, htmlBuffer, HTML_BUFFER_SIZE);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```

int ticketIDStartPosition = HTMLSource.indexOf("ticket=") + 7;
String ticketID = HTMLSource.substring(ticketIDStartPosition, HTMLSource.indexOf("\"", ticketIDStartPosition));
reader_portal.close();
//////////////////// HKU portal //////////////////////////////////////
//////////////////// Moodle //////////////////////////////////////
// URL url_moodle = new URL("http://moodle.hku.hk/login/index.php?authCAS=CAS&ticket=" + ticketID);
URL url_moodle = new URL("https://moodle.hku.hk/login/index.php?authCAS=CAS&ticket=" + ticketID);
conn_moodle = url_moodle.openConnection();
((URLConnection) conn_moodle).setInstanceFollowRedirects(true);
BufferedReader reader_moodle = new BufferedReader(new InputStreamReader(conn_moodle.getInputStream()));
/// handling redirects to HTTPS protocol
while (true) {
    String redirect_moodle = conn_moodle.getHeaderField("Location");
    if (redirect_moodle != null) {
        URL new_url_moodle = new URL(url_moodle, redirect_moodle);
        if (moodle_conn_type == HTTPCONNECTION_TYPE) {
            ((URLConnection) conn_moodle).disconnect();
        } else {
            ((HttpsURLConnection) conn_moodle).disconnect();
        }
        conn_moodle = new_url_moodle.openConnection();
        if (new_url_moodle.getProtocol().equals("http")) {
            moodle_conn_type = HTTPCONNECTION_TYPE;
            ((URLConnection) conn_moodle).setInstanceFollowRedirects(true);
        } else {
            moodle_conn_type = HTTPSCONNECTION_TYPE;
            ((HttpsURLConnection) conn_moodle).setInstanceFollowRedirects(true);
        }
        url_moodle = new_url_moodle;
        //String cookie = conn_moodle.getHeaderField("Set-Cookie");
        //if (cookie != null) {
        // conn_moodle2.setRequestProperty("Cookie", cookie);
        //}
        reader_moodle = new BufferedReader(new InputStreamReader(conn_moodle.getInputStream()));
    } else {
        break;
    }
}
HTMLSource = ReadBufferedHTML(reader_moodle, htmlBuffer, HTML_BUFFER_SIZE);
reader_moodle.close();
return HTMLSource;
//////////////////// Moodle //////////////////////////////////////
} catch (Exception e) {
    return "Fail to login";
} finally {
    // When HttpClient instance is no longer needed,
    // shut down the connection manager to ensure
    // immediate deallocation of all system resources
    if (conn_portal != null) {
        conn_portal.disconnect();
    }
    if (conn_moodle != null) {
        if (moodle_conn_type == HTTPCONNECTION_TYPE) {
            ((URLConnection) conn_moodle).disconnect();
        } else {
            ((HttpsURLConnection) conn_moodle).disconnect();
        }
    }
}
}
}

```

### 3.3 Add the following function to accept all the HTTPS certificates

```
// trusting all certificate
public void doTrustToCertificates() {
    TrustManager[] trustAllCerts = new TrustManager[]{
        new X509TrustManager() {
            public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                return null;
            }

            public void checkClientTrusted(java.security.cert.X509Certificate[] certs, String authType) {
            }

            public void checkServerTrusted(java.security.cert.X509Certificate[] certs, String authType) {
            }
        }
    };
    try {
        // Install the all-trusting trust manager
        SSLContext sc = SSLContext.getInstance("SSL");
        sc.init(null, trustAllCerts, new java.security.SecureRandom());
        HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Note: If you watch the video, there was an extra “}” in the previous pdf.

### 3.4 Initialize default cookie manager, and calling “doTrustToCertificates()” in the onCreate() function at MainActivity Class

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btn_Login = (Button) findViewById(R.id.btn_Login);
    txt_UserName = (EditText) findViewById(R.id.txt_UserName);
    txt_UserPW = (EditText) findViewById(R.id.txt_UserPW);

    // Register the Login button to click listener
    // Whenever the button is clicked, onClick is called
    btn_Login.setOnClickListener(this);

    doTrustToCertificates();
    CookieHandler.setDefault(new CookieManager());
}
```

3.5 Browse to app ⑦ manifests ⑦ AndroidManifest.xml, and adding permission for accessing internet.

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

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="hk.hku.cs.ws1" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="WS1"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="WS1" >
            <intent-filter>
```

#### 4 Alert Box, Regular Expression, Loading Screen, AsyncTask

4.1 Add the following functions into **MainActivity** class

##### 4.1.1 Alert the user if the connection failed

```
protected void alert(String title, String mymessage) {
    new AlertDialog.Builder(this)
        .setMessage(mymessage)
        .setTitle(title)
        .setCancelable(true)
        .setNegativeButton(android.R.string.cancel,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                }
            }
        )
        .show();
}
```

##### 4.1.2 Parse HTML source with Regular Expression and switch activity About

Regular Expression:

[http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression)

Java provides the java.util.regex package for pattern matching with regular expressions. Java regular expressions are very similar to the Perl programming language and very easy to learn. A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. They can be used to search, edit, or manipulate text and data. The java.util.regex package primarily consists of the following three classes:

- ★ **Pattern Class** – A Pattern object is a compiled representation of a regular expression. The Pattern class provides no public constructors. To create a pattern, you must first invoke one of its public static compile() methods, which will then return a Pattern object. These methods accept a regular expression as the first argument.

- ★ **Matcher Class** – A `Matcher` object is the engine that interprets the pattern and performs match operations against an input string. Like the `Pattern` class, `Matcher` defines no public constructors. You obtain a `Matcher` object by invoking the `matcher()` method on a `Pattern` object.
- ★ **`PatternSyntaxException`** – A `PatternSyntaxException` object is an unchecked exception that indicates a syntax error in a regular expression pattern.

```

public void parse_HTML_Source_and_Switch_Activity(String HTMLsource) {
    Pattern p_coursename = Pattern.compile("<h3 class=\\\"coursename\\\".*?>.*?</a>");
    Matcher m_course = p_coursename.matcher(HTMLsource);
    Pattern p_teachercandidates = Pattern.compile("<div class=\\\"teachers\\\">Teacher: <.*?>(<.*?></a>");
    Matcher m_teachercandidates = p_teachercandidates.matcher(HTMLsource);

    ArrayList<String> cname = new ArrayList<String>();
    ArrayList<String> cteachers = new ArrayList<String>();
    ArrayList<String> cteachersfinal = new ArrayList<String>();
    ArrayList<Integer> cnamePos = new ArrayList<Integer>();
    ArrayList<Integer> cteachersPos = new ArrayList<Integer>();
    ArrayList<Integer> cteachersIdx = new ArrayList<Integer>();

    while (m_course.find()) {
        String course_name = m_course.group(1);
        Integer pos = m_course.start();
        boolean flag = true;
        for (String sss : cname) {
            if (sss.equals(course_name)) {
                flag = false;
            }
        }
        if (flag) {
            cname.add(course_name);
            cnamePos.add(pos);
        }
    }

    while (m_teachercandidates.find()) {
        String string_teachername = m_teachercandidates.group(1);
        // int nameStartPosition = string_teachername.indexOf(">")+1;
        // int nameEndPosition = string_teachername.indexOf("</a>");
        // String teacher_name = string_teachername.substring(nameStartPosition, nameEndPosition);
        cteachers.add(string_teachername);
        Integer pos = m_teachercandidates.start();
        cteachersPos.add(pos);
    }

    Intent intent = new Intent(getBaseContext(), CourseListActivity.class);

    int cldx = 0;
    for (int i = 0; i < cteachersPos.size(); ) {
        int cpos0 = -1, cpos1 = -1;
        int tpos = cteachersPos.get(i);
        if (cldx < cnamePos.size()) {
            cpos0 = cnamePos.get(cldx);
        }
        if (cldx + 1 < cnamePos.size()) {
            cpos1 = cnamePos.get(cldx + 1);
        }
        if (cpos0 < 0 || tpos < cpos0) { /// a course with 2 teachers!? Assume the teacher belongs to the previous course
            cteachersIdx.add(cldx - 1);
            i++;
        } else if (cpos1 < 0 || (cpos0 < tpos && cpos1 > tpos)) {
            cteachersIdx.add(cldx);
            i++;
            cldx++;
        } else { /// tpos > cpos1 ==> teacher belongs to next classes
            cldx++;
        }
    }
}

```



```
    }  
  }  
  for (int i = 0; i < cname.size(); i++) {  
    String tname = "";  
    for (int j = 0; j < cteachersidx.size(); j++) {  
      int cidx = cteachersidx.get(j);  
      if (cidx == i) {  
        tname += cteachers.get(j);  
      }  
    }  
    cteachersfinal.add(tname);  
  }  
  intent.putStringArrayListExtra("CourseName", cname);  
  intent.putStringArrayListExtra("Teachers", cteachersfinal);  
  startActivity(intent);  
}
```

### 4.1.3 Loading Screen and AsyncTask

```
public void connect( final String userName, final String userPW ){
    final ProgressDialog pdialog = new ProgressDialog(this);
    pdialog.setCancelable(false);
    pdialog.setMessage("Logging in ...");
    pdialog.show();
    AsyncTask<String, Void, String> task = new AsyncTask<String, Void, String>() {
        boolean success;
        String moodlePageContent;
        @Override
        protected String doInBackground(String... arg0) {
            // TODO Auto-generated method stub
            success = true;
            moodlePageContent = getMoodleFirstPage(userName, userPW);
            if( moodlePageContent.equals("Fail to login") )
                success = false;
            return null;
        }
        @Override
        protected void onPostExecute(String result) {
            if (success) {
                parse_HTML_Source_and_Switch_Activity( moodlePageContent );
            } else {
                alert( "Error", "Fail to login" );
            }
            pdialog.hide();
        }
    }.execute("");
}
```

## AsyncTask

Added in API level 3

extends [Object](#)

[java.lang.Object](#)

↳ [android.os.AsyncTask<Params, Progress, Result>](#)

## Class Overview

**AsyncTask** enables proper and easy use of the UI thread. This class allows to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers.

**AsyncTask** is designed to be a helper class around [Thread](#) and [Handler](#) and does not constitute a generic threading framework. **AsyncTasks** should ideally be used for short operations (a few seconds at the most.) If you need to keep threads running for long periods of time, it is highly recommended you use the various APIs provided by the [java.util.concurrent](#) package such as [Executor](#), [ThreadPoolExecutor](#) and [FutureTask](#).

An asynchronous task is defined by a computation that runs on a background thread and whose result is published on the UI thread. An asynchronous task is defined by 3 generic types, called [Params](#), [Progress](#) and [Result](#), and 4 steps, called [onPreExecute](#), [doInBackground](#), [onProgressUpdate](#) and [onPostExecute](#).

## 5 Modifying **onClick** function

```
@Override
public void onClick(View v) {
    // TODO Auto-generated method stub
    if (v.getId() == R.id.btn_Login) {
        String uname = txt_UserName.getText().toString();
        String upassword = txt_UserPW.getText().toString();

        connect( uname, upassword );
    }
}
```

6 At this point, the Moodle app has been completed!

7 Note: the “asynctask” is deprecated, but the program still works. The connect function can be replaced with the code below.

```
public void connect( final String userName, final String userPW ){

    final ProgressDialog pdialog = new ProgressDialog(this);
    pdialog.setCancelable(false);
    pdialog.setMessage("Logging in ...");
    pdialog.show();

    //Reference: Modified the AsyncTask using this website
    //https://stackoverflow.com/questions/58767733/android-async-task-api-deprecating-in-android-11-what-are-the-alternatives
    ExecutorService executor = Executors.newSingleThreadExecutor();

    Handler handler = new Handler(Looper.getMainLooper());

    executor.execute(new Runnable() {

        boolean success;

        String moodlePageContent;

        @Override
        public void run() {

            success =true;

            moodlePageContent =

                getMoodleFirstPage(userName, userPW);

            if(moodlePageContent.equals("Fail to login"))

                success =false;

            handler.post(new

                Runnable() {
```

```

        @Override

        public void run () {

            if (success) {

                parse_HTML_Source_and_Switch_Activity(moodlePageContent);

            } else {

                alert("Error", "Fail to login");

            }

            pdialog.hide();

        }

    });

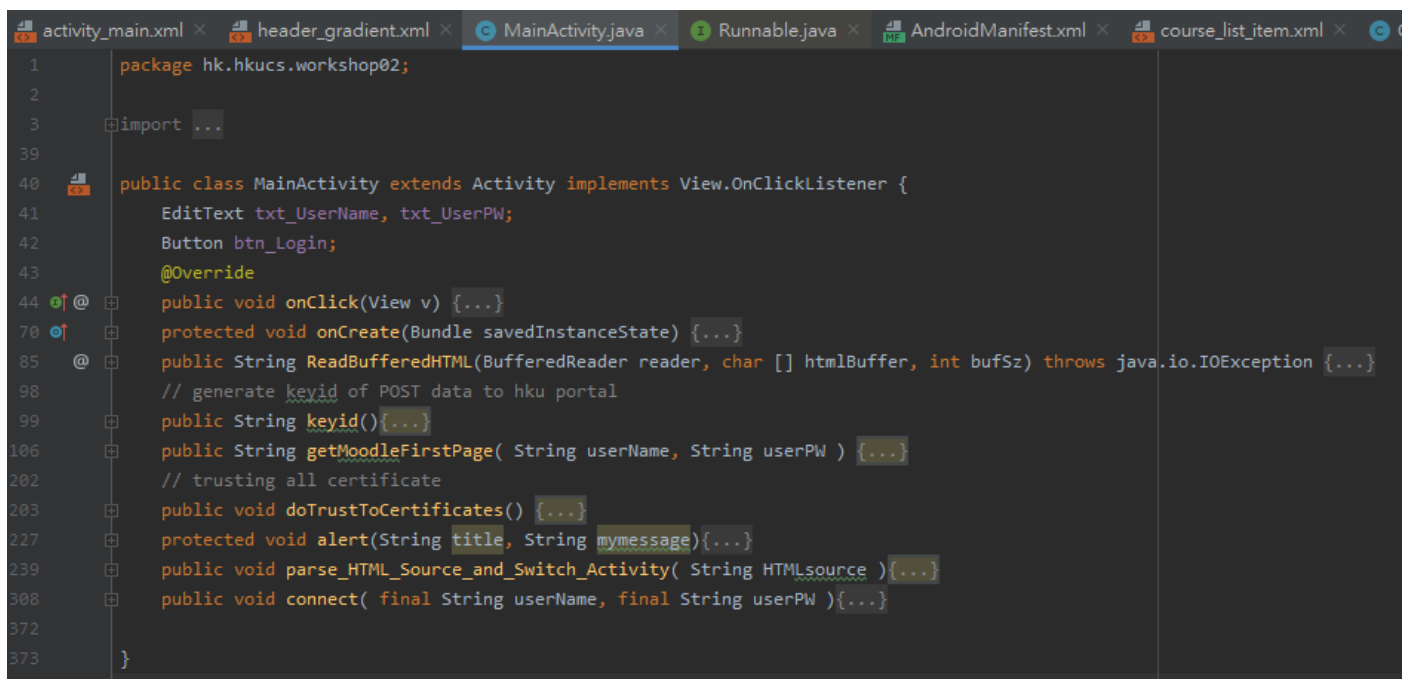
}

});

}

```

## 8 Here is the summary of the MainActivity.java



```

1  package hk.hkucs.workshop02;
2
3  import ...
39
40  public class MainActivity extends Activity implements View.OnClickListener {
41      EditText txt_UserName, txt_UserPW;
42      Button btn_Login;
43      @Override
44      public void onClick(View v) {...}
70
71      protected void onCreate(Bundle savedInstanceState) {...}
85
86      public String ReadBufferedHTML(BufferedReader reader, char [] htmlBuffer, int bufSz) throws java.io.IOException {...}
98      // generate keyid of POST data to hku portal
99      public String getKeyid(){...}
106     public String getMoodleFirstPage( String userName, String userPW ) {...}
202     // trusting all certificate
203     public void doTrustToCertificates() {...}
227     protected void alert(String title, String mymessage){...}
239     public void parse_HTML_Source_and_Switch_Activity( String HTMLsource ) {...}
308     public void connect( final String userName, final String userPW ) {...}
372
373 }

```

9 If you need to auto indent the program, Just Press **CTRL+ALT+L** on Windows or **Command+Option+L** on Mac.

Short cut Reference: <https://medium.com/mindorks/11-android-studio-shortcuts-every-android-developer-must-know-a153e736e611#:~:text=Again%20this%20is%20one%20of,%2BOption%2BL%20on%20Mac.>

**Please save your work, zip the project folder and submit it to Moodle as a proof of workshop participation.**