**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.

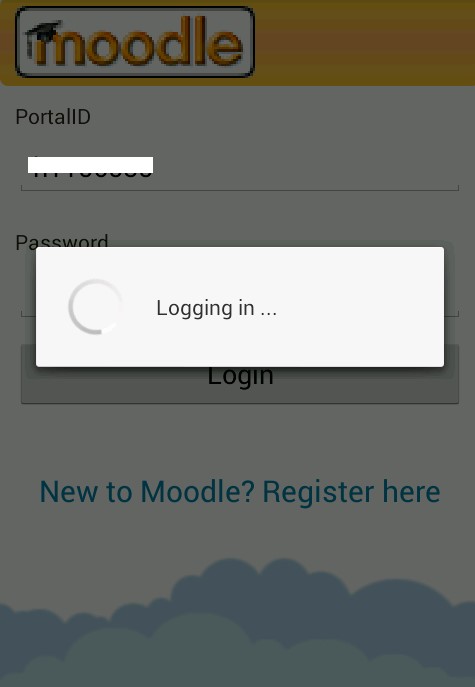
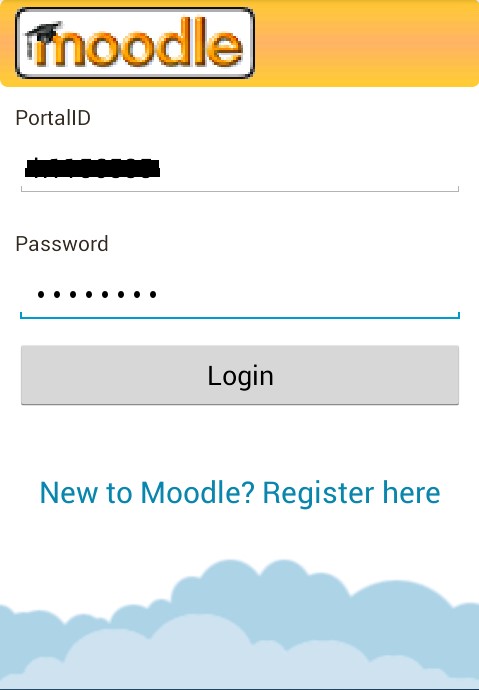
Now, we are going to get real data from **HKU Moodle** by using **HTTP** requests.

C

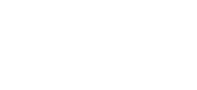
odes will be provided

for reference

!



HKU



Portal ID

Portal PW



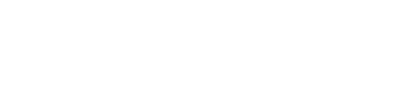
HTML source of moodle



Parser (Regular Expression)



HTML source of moodle



ArrayList<String> CourseName

ArrayList<String> Teachers

**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.

1. Importing the project of workshop 2

1. Login to hkuportal, and redirect to moodle

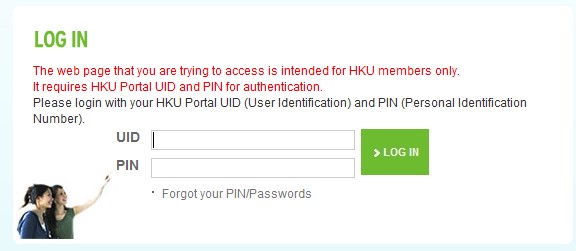
3.1 Introduction

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.

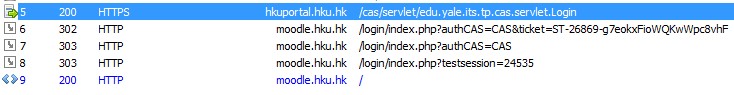
The flow chart of login moodle: Step 1: http://moodle.hku.hk



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



**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=" + keyid() +

"&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); |

|  |
| --- |
| 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();  ((HttpURLConnection) 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) {  ((HttpURLConnection) 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;  ((HttpURLConnection) 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) {  ((HttpURLConnection) 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 → manifests → AndroidManifest.xml, and adding permission for accessing internet.

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



1. Alert Box, Regular Expression, Loading Screen, AsyncTask
   1. Add the following functions into **MainActivity** class
      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();  } |

* + 1. Parse HTML source with Regular Expression and switch activity About Regular Expression:

http://en.wikipedia.org/wiki/Regular\_expression

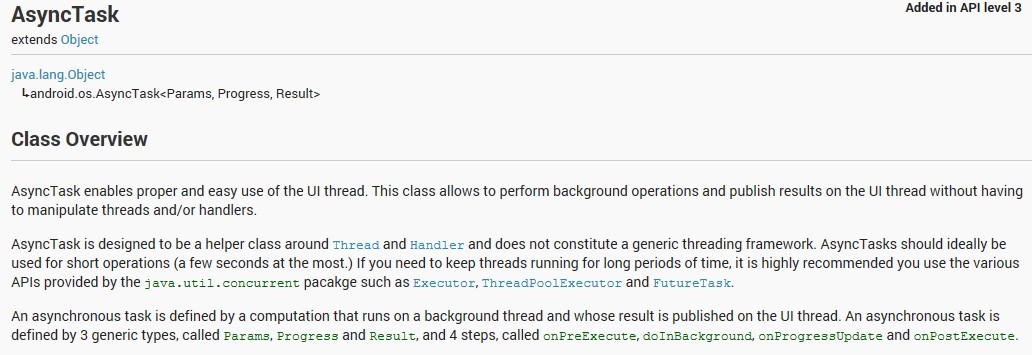
Java provides the java.util.regex package for pattern matching with regular expressions. Java regular expressions are very similar to the Perl programming language and very easy to learn. A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, 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 cIdx = 0;  for (int i = 0; i < cteachersPos.size(); ) {  int cpos0 = -1, cpos1 = -1;  int tpos = cteachersPos.get(i);  if (cIdx < cnamePos.size()) {  cpos0 = cnamePos.get(cIdx);  }  if (cIdx + 1 < cnamePos.size()) {  cpos1 = cnamePos.get(cIdx + 1);  }  if (cpos0 < 0 || tpos < cpos0) { /// a course with 2 teachers!? Assume the teacher belongs to the previous course  cteachersIdx.add(cIdx - 1);  i++;  } else if (cpos1 < 0 || (cpos0 < tpos && cpos1 > tpos)) {  cteachersIdx.add(cIdx);  i++;  cIdx++;  } else { /// tpos > cpos1 ==> teacher belongs to next classes  cIdx++; |
| }  }  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);  } |

* + 1. 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("");  } |



1. Modifying onClick function

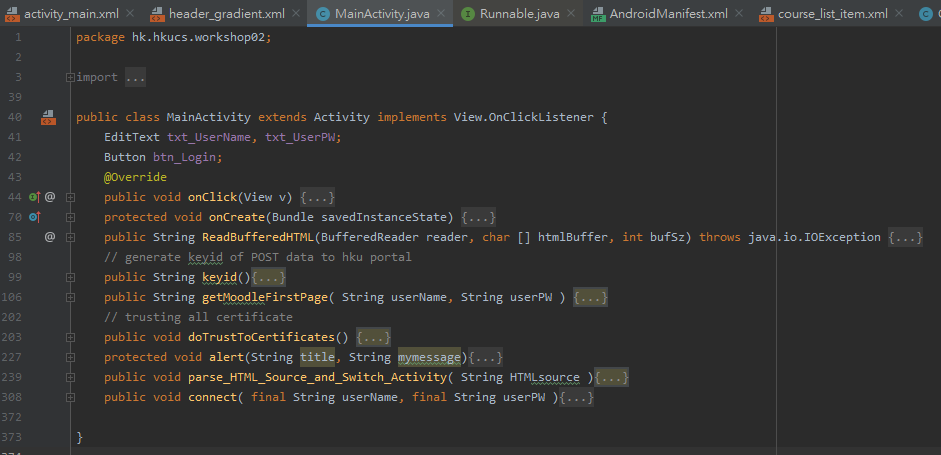
|  |
| --- |
| @Override  public void onClick(View v) {  // TODO Auto-generated method stub if (v.getId() == R.id.btn\_Login) {  String uname = txt\_UserName.getText().toString();  String upassword = txt\_UserPW.getText().toString();    connect( uname, upassword );  }  } |

1. 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-asynctask-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



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.**