Software Engineering - M1
TP 1: Android basics

J.-F. Lalande

jean-francois.lalande@insa-cvl.fr

1 My first Android application

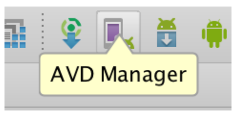
We will use Android Studio to develop applications. You can use your own version (on your computer) or the installed version on the school's computers.

1.1 Discovering the Android Studio environment

Exercise 1 Launch Android Studio and generate a new "Hello World" Android Project with the wizard (API 23) with a custom package name (e.g. myname.TP1).

Exercise 2 Connect your phone using a micro USB cable. In case you have no phone, check the AVD manager by clicking the icon below. Run the emulator if it is already configured. Do not close this emulator later : it will be used and reused to run your app.

Exercise 3 Run your application (green triangle). If no AVD or real phone is found, Android Studio will propose you to run a new emulator. You should get your application pushed and the first activty "hello world !" displayed.



1.2 Resources

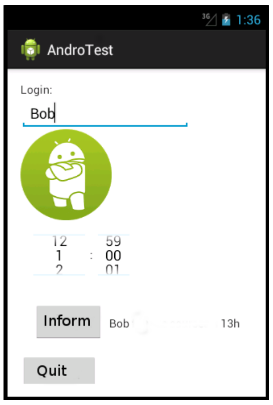
Exercise 4 Modify a resource "values", for example the string "hello_world". Look at the resulting XML and run your app again.

Exercise 5 Drop down an image (using your file explorer) in the drawable directory. Open the R class[1] that has been generated and check the generated id of your image.

1.3 Graphical elements

Exercise 6 Create a first graphical interface in your main activity. Use the layout assistant. Use a vertical LinearLayout as the root graphical element (and not a RelativeLayout). Then add :

- a text view containing "Login :"
- a text field
- an image
- an object for choosing time : TimePicker (use the attribute timePickerMode="spinner" to have a more convenient one)
- a LinearLayout with horizontal orientation with inside :
  - a button on the left



1. You can find R.java by typing "R" in your activity, then right click > Go To > Implementation.

  - — a text view on the right
- — a button "Inform" that does nothing for now
- — a last button to quit

1.4 Events

For the moment, buttons do nothing. We should add adequate code for handling events.

Exercise 7 Check your XML layout for putting relevant ids for your graphical elements.

Exercise 8 When clicking on the first button, copy the login text and put it in the right text view. Add to the text the time to get something like "JFL - 22h11".

Exercise 9 Code the Quit button.

1.5 Intent and activities

Exercise 10 Create a new activity Authentication. This activity should be called when the button "Inform" is hit. Send the login and the time to this new activity using extras.

1.6 Authentication activity design

Exercise 11 On the activity Authentication, display the login of the user.

Exercise 12 Add, below, an EditText for entering the password.

Exercise 13 Create a button "Authenticate" for starting the authentication.

Exercise 14 Look at the post and its answer at `http://stackoverflow.com/q/18021148/1156363` and code an overlay progress bar that is displayed when the button "Authenticate" is hit. You need to adapt your layout to your needs. In particular, you see that the `<FrameLayout android:id="@+id/progressBarHolder"/>` is invisble, by default, and appears when the AsynTask starts.

1.7 Handle the authentication

Exercise 15 Create a Java class MyCredential that holds the login and the password of the user.

Exercise 16 Modify your ASyncTask to pass an object of type MyCredential. This way, the method doInBackround will be `protected Long doInBackground(MyCredential... mycreds)`. Now we are ready to authenticate the user.

Exercise 17 Perform an authentication of the user using the url `http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/dev/service.php?login=xxxx&password=yyyy` and the class HttpGet request. The parameters of this service are : login and password. To authenticate, the password should be the length of the login string. Do not forget to add the INTERNET permission. Note that service.php is slow : this is why doing the authentication process in an AsynTask is better :)

Exercise 18 If the authentication fails, display the result in a TextView in the Authentication activity. If it is successful, launch a third activity.

2 Broadcasting information

When authenticating, we propose to log the attempt of authentication in another application. This way, the other app will record and display all the connection attempts.

2.1 Getting the attempt connection information

Exercise 19 When the authentication fails or succeeds, send a informative broadcast with the name "andro.jf.broadcast.login" and several keys/values :

— success : yes/no
— login : the login value
— time : time of the authentication

Exercise 20 Create a new Android Studio project named AndroTestReceiver containing a "blank activity".

Exercise 21 In the AndroTestReceiver app, in the Manifest, add a receiver declaration that filters the intent "andro.jf.broadcast".

Exercise 22 Create the class *MyBroadcastReceiver* that should extend *BroadcastReceiver*. Overwrite the method *onReceive(Context context, Intent intent)* with the following code :

```
Bundle extra = intent.getExtras();
if (extra != null) {
  String val = extra.getString("message le code");
  Toast.makeText(context, "Broadcast message received: " + val, Toast.LENGTH_SHORT).show();
}
```

Exercise 23 Push your second application in the emulator and test the broadcast receiver from the first app.

2.2 Storing connection information

For now, the second application displays a Toast : this is just a way to debug and verify that the intent is received. But, if we want to store the connection attempt, we need to create a database and use it for displaying data.

Exercise 24 Look at the course about SQL and also the Android dev page at `http://developer.android.com/training/basics/data-storage/databases.html`. Recreate the class FeedReaderDbHelper for handling your database. Adapt the columns of the database by adjusting *`SQL_CREATE_ENTRIES`* : the columns should store the 3 informations success, login, time.

Exercise 25 In MyBroadcastReceiver, when you receive the intent, enter a row of data in the database.

2.3 Displaying the database data

Exercise 26 Create a ListView in the second application layout. This ListView will be populated with the content of the database. Create a layout file itemListLayout.xml that will be used for displaying an item of the list. This layout will display at least 3 TextView "success or failure", "login" and the time of authentication attempt.

Exercise 27 Look at the course about the commplex list view. You should customize the generic ArrayAdapter for example with ArrayAdapter<MyData>. The MyData object will contain the three strings to display. Go through your database and extract the data and add the data to the ArrayAdapter. This way you should succeed to display the authentication attemps in your list :

```
------------------------------
Attempt: failure | toto | 12h24
------------------------------
Attempt: failure | titi | 12h26
------------------------------
Attempt: success | jfl  | 12h36
------------------------------
```