Activity Life Cycle (and Crtl+F11, and telnet to emulator)

You have coded a few examples now and you may start wondering why is the method into which we have been writing most of the code named "onCreate".

Do you remember that your application may be composed of multiple activities but that only one is fully active at a time? What happens when your application moves from one activity to the next? And back?

**Important and mandatory:** Read about [the activity life cycle in the Android documentation](http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle) to discover the answers to these questions.

Now let's experiment a bit with this life cycle by "tracing" an application as it executes the onCreate, onStart, onRestart, onResume, onPause, onStop and onDestroy methods:

1. create a new Android Studio project with our usual choices (API 16, Empty Activity, generation of MainActivity.java and activity\_main.xml))

2. modify the MainActivity to display a pop-up message each time the activity goes through one of the life cycle related methods:

package fr.centralesupelec.galtier.lifecycle;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toast.makeText(getApplicationContext(), "onCreate", Toast.LENGTH\_SHORT

          ).show();

}

@Override

protected void onStart() {

super.onStart();

Toast.makeText(getApplicationContext(),"onStart", Toast.LENGTH\_SHORT

          ).show();

}

@Override

protected void onRestart() {

super.onRestart();

Toast.makeText(getApplicationContext(),"onRestart", Toast.LENGTH\_SHORT

          ).show();

}

@Override

protected void onResume() {

super.onResume();

Toast.makeText(getApplicationContext(),"onResume", Toast.LENGTH\_SHORT

          ).show();

}

@Override

protected void onPause() {

Toast.makeText(getApplicationContext(),"onPause", Toast.LENGTH\_SHORT

          ).show();

super.onPause();

}

@Override

protected void onStop() {

Toast.makeText(getApplicationContext(),"onStop", Toast.LENGTH\_SHORT

          ).show();

super.onStop();

}

@Override

protected void onDestroy() {

Toast.makeText(getApplicationContext(),"onDestroy", Toast.LENGTH\_SHORT

          ).show();

super.onDestroy();

}

}

3. execute the app on the emulator. You should see the sequence *onCreate - onStart - onResume.*

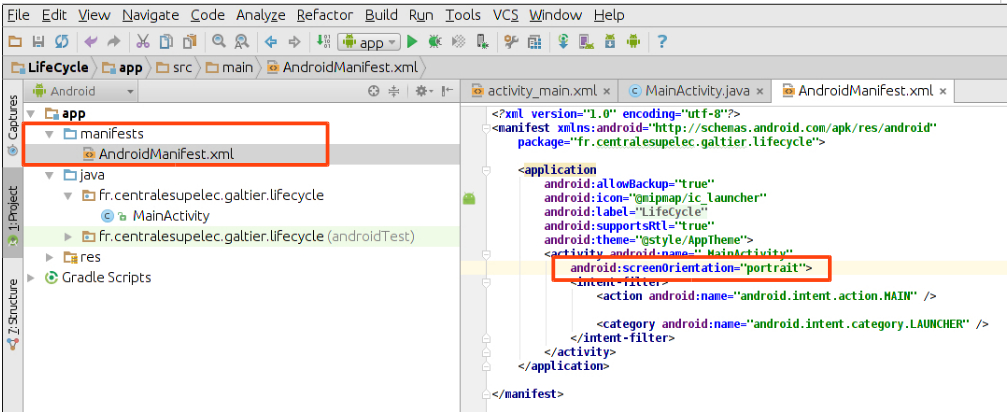
4. experiment with the following scenarios:

- click the "back" button (ie the left one): you should see *onPause - onStop - onDestroy*, and next return to the app: you should see the sequence *onCreate - onStart - onResume* again.

- click the "home" button (ie the center one): you should see *onPause - onStop*, and next return to the app: you should see *onRestart - on Start - onResume*.

- press "Crtl + F11" to rotate the emulator: you should see *onPause - onStop - onDestroy - onCreate - onStart - onResume*. Press "Crtl + F11 to return to a portrait orientation.

- edit the AndroidManifest.xml file to prevent the rotation of the screen:



<?xml version="1.0" encoding="utf-8"?>

<manifest ...>

<application ...>

<activity ...

android:screenOrientation="portrait">

...

</activity>

</application>

</manifest>

Press Crtl+F11 again: the activity is no longer destroyed and re-created.

- Now we will simulate an incoming phone call. In order to do that, you need to telnet to the emulator.

If you are working with Linux or Mac, telnet is most probably already installed. Open a terminal and type "telnet localhost 5554".

If you are working with Windows, install the putty software (from [putty.org](http://www.putty.org/)) and run it. In the "Host Name" field enter "localhost", in the "Port" field enter "5554" and choose the "Telnet" radio button as a "Protocol", then click "Open". If you get the message "KO: unknown command, try 'help'" it may be because telnet is not installed or activated on your system. For Window 7, follow [the steps described here](http://www.itdoescompute.com/2009/10/29/how-to-enable-telnet-in-windows-7/); or try choosing the "raw" protocol radio button instead of "telnet".

In the telnet session, type "gsm call 12345679" and press Enter. A call notification appears at the top of the screen, and partially covers the current activity. Select "Dismiss": the notification should disappear and no pop-up should be displayed.

Simulate a new incoming call (type "gsm call 123456789" again in the telnet session). This time accept the call (select "answer"). You should see the *onPause* pop-up and the call activity completely covers your activity. Now end the call: the call activity disappears and the screen returns to your activity with the *onResume* pop-up.