

TP - ANDROID

Threads & Handlers

Android permet d'utiliser les Threads de Java. Cependant, un thread en background doit respecter certaines règles, telles que ne pas changer d'éléments dans l'interface utilisateur en foreground pour des raisons d'accès concurrents. Pour cela, il doit passer par la classe Handler ou par les AsyncTask.

Exemple Handler

Créer un projet "fr.efrei.android.moi.handler" avec l'activité "ProgressTestActivity".

Créer le layout suivant (main.xml) :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ProgressBar android:id="@+id/progressBar1"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:indeterminate="false"
        android:max="10" android:padding="4dip"></ProgressBar>
    <Button android:text="Start Progress"
        android:onClick="startProgress"
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"></Button>
</LinearLayout>
```

Changez l'activité de la façon suivante :

```
public class ProgressTestActivity extends Activity {
    private Handler handler;
    private ProgressBar progress;
```

```
/** Appelé quand l'activité est créée en premier. */
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    progress = (ProgressBar) findViewById(R.id.progressBar1);
    handler = new Handler();
}

public void startProgress(View view) {
    // Faire quelque chose de long !
    Runnable runnable = new Runnable() {
        @Override
        public void run() {
            for (int i = 0; i <= 10; i++) {
                final int value = i;
                try {
                    Thread.sleep(2000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        progress.setProgress(value);
                    }
                });
            }
        }
    };
    new Thread(runnable).start();
}
}

```

Lancez votre application. Quand vous pressez sur le bouton, la barre de progression sera mise à jour. A tester et à comprendre !

Exemple Async Task

On va utiliser dans cet exemple une tâche asynchrone pour télécharger le contenu d'une page Web.

Créer un nouveau projet « fr.efrei.android.moi.asyncTask » avec l'activité « ReadWebpageAsyncTask ». Ajouter la permission « android.permission.INTERNET » dans le manifest. Créer le layout suivant.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
    >

    <Button android:layout_height="wrap_content"
        android:layout_width="match_parent" android:id="@+id/readWebpage"
        android:onClick="readWebpage" android:text="Load Webpage"></Button>
    <TextView android:id="@+id/TextView01" android:layout_width="match_parent"
        android:layout_height="match_parent" android:text="Example
    Text"></TextView>
</LinearLayout>

```

Et voilà le code ! Regardez à nouveau les définitions des différentes classes dans les packages Java de la jdk + les nouvelles classes dans le sdk Android.

```

package fr.efrei.android.moi.asynctask;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;

import org.apache.http.HttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.app.Activity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class ReadWebpageAsyncTask extends Activity {
    private TextView textView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        textView = (TextView) findViewById(R.id.TextView01);
    }

    private class DownloadWebPageTask extends AsyncTask<String, Void,
String> {
        @Override
        protected String doInBackground(String... urls) {
            String response = "";
            for (String url : urls) {
                DefaultHttpClient client = new
DefaultHttpClient();
                HttpGet httpGet = new HttpGet(url);
                try {
                    HttpResponse execute =
client.execute(httpGet);

```

```

        execute.getEntity().getContent();

        InputStream content =

        BufferedReader buffer = new
                                new
        InputStreamReader(content));

        String s = "";
        while ((s = buffer.readLine()) !=
                                response += s;
        }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return response;
}

@Override
protected void onPostExecute(String result) {
    textView.setText(result);
}

}

public void readWebpage(View view) {
    DownloadWebPageTask task = new DownloadWebPageTask();
    task.execute(new String[] { "http://www.efrei.fr" });
}
}

```

Cycle de vie Activity & Thread

Créer un projet. Changez le manifest de la façon suivante :

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.efrei.android.moi.threads lifecycle"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission android:name="android.permission.INTERNET"></uses-
permission>

    <application android:icon="@drawable/icon"
    android:label="@string/app_name">
        <activity android:name=".ThreadsLifecycleActivity"

```

```

        android:label="@string/app_name">
        <intent-filter>
            <action
android:name="android.intent.action.MAIN" />
            <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

</application>
</manifest>

```

Changer le « main.xml » aussi !

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:layout_height="wrap_content"
        android:layout_width="match_parent"
android:id="@+id/linearLayout1">
        <Button android:onClick="downloadPicture"
            android:layout_height="wrap_content"
android:text="Click to start download"
            android:layout_width="wrap_content"></Button>
        <Button android:onClick="resetPicture"
            android:layout_height="wrap_content"
            android:text="Reset Picture"
            android:layout_width="wrap_content"></Button>
    </LinearLayout>
    <ImageView android:src="@drawable/icon"
android:id="@+id/imageView1"
        android:layout_height="match_parent"
android:layout_width="match_parent"></ImageView>
</LinearLayout>

```

Et voilà le code de l'application qui va télécharger une image et en même temps avoir une boîte de dialogue jusqu'à ce que le téléchargement soit terminé !

```

package fr.efreii.android.threads lifecycle;

import java.io.IOException;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.StatusLine;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;

```

```

import org.apache.http.client.methods.HttpUriRequest;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;

import android.app.Activity;
import android.app.ProgressDialog;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.ImageView;

public class ThreadsLifecycleActivity extends Activity {

    private static ProgressDialog dialog;
    private static ImageView imageView;
    private static Bitmap downloadBitmap;
    private static Handler handler;
    private Thread downloadThread;

    /** Appelé quand une activité est créée. */

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // handler pour MAJ de l'UI
        handler = new Handler();
        // récupérer l'imageView la plus récente après le restart de
l'application
        imageView = (ImageView) findViewById(R.id.imageView1);
        // Est-ce que l'image a été téléchargée?
        if (downloadBitmap != null) {
            imageView.setImageBitmap(downloadBitmap);
        }
        // Vérifier si le thread tourne déjà
        downloadThread = (Thread) getLastNonConfigurationInstance();
        if (downloadThread != null && downloadThread.isAlive()) {
            dialog = ProgressDialog.show(this, "Download",
"downloading");
        }
    }

    public void resetPicture(View view) {
        if (downloadBitmap != null) {
            downloadBitmap = null;
        }
        imageView.setImageResource(R.drawable.icon);
    }

    public void downloadPicture(View view) {
        dialog = ProgressDialog.show(this, "Download",
"downloading");
        downloadThread = new MyThread();
        downloadThread.start();
    }

    // Sauvegarder le thread

```

```

@Override
public Object onRetainNonConfigurationInstance() {
    return downloadThread;
}

// supprimer le dialog si l'activité est détruite
@Override
protected void onDestroy() {
    if (dialog != null && dialog.isShowing()) {
        dialog.dismiss();
        dialog = null;
    }
    super.onDestroy();
}

// method servant à télécharger l'image
static private Bitmap downloadBitmap(String url) throws IOException
{
    HttpUriRequest request = new HttpGet(url.toString());
    HttpClient httpClient = new DefaultHttpClient();
    HttpResponse response = httpClient.execute(request);

    StatusLine statusLine = response.getStatusLine();
    int statusCode = statusLine.getStatusCode();
    if (statusCode == 200) {
        HttpEntity entity = response.getEntity();
        byte[] bytes = EntityUtils.toByteArray(entity);

        Bitmap bitmap = BitmapFactory.decodeByteArray(bytes,
0,
        bytes.length);
        return bitmap;
    } else {
        throw new IOException("Download failed, HTTP
response code "
        + statusCode + " - " +
statusLine.getReasonPhrase());
    }
}

static public class MyThread extends Thread {
    @Override
    public void run() {
        try {
            // On simule un réseau lent
            try {
                new Thread().sleep(5000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            downloadBitmap =
downloadBitmap("http://www.efrei.fr/sites/default/files/Homepage2.jpg?13161
92442");
            handler.post(new MyRunnable());
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
        }
    }
}
}

```

```
static public class MyRunnable implements Runnable {  
    public void run() {  
        imageView.setImageBitmap(downloadBitmap);  
        dialog.dismiss();  
    }  
}  
}
```

Lancer l'application et taper sur le bouton pour lancer le téléchargement. Vous pouvez tester la correction du cycle de vie via l'émulateur avec la touche CTRL+F11 qui change l'orientation.