Utilisation d’un handler

# Démonstration 3 du module 6

|  |
| --- |
| Cette démonstration montre comment utiliser les handler. La fonction onClickAvecHandler contient le code utilisant les handlers. |

Déroulement

* Créer un nouveau projet avec une activité nommée MainActivity de type « Navigation Drawer activity ».

|  |  |
| --- | --- |
| Fichier | Description |
| Activity\_main.java | Fichier contenant le code p |
| main\_activity.xml | Fichier représentant l’IHM de l’activité. |

* Modifier le contenu de main\_activity.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="fr.eni.demonstrationm07d01.MainActivity">  
  
 <ProgressBar  
 android:id="@+id/pb\_demonstration"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:visibility="visible"  
 android:max="10"  
 android:progress="0"  
 />  
  
 <Button  
 android:id="@+id/btn\_sans\_thread"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/pb\_demonstration"  
 android:text="GO sans Thread"  
 android:onClick="onClickSansThread"  
 />  
  
 <Button  
 android:id="@+id/btn\_avec\_thread"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/btn\_sans\_thread"  
 android:text="GO avec Thread"  
 android:onClick="onClickAvecThread"  
 />  
  
 <Button  
 android:id="@+id/btn\_avec\_asynctask"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/btn\_avec\_thread"  
 android:text="GO avec AsyncTask"  
 android:onClick="onClickAvecASyncTask"  
 />  
  
 <Button  
 android:id="@+id/btn\_avec\_handler"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/btn\_avec\_asynctask"  
 android:text="GO avec Handler"  
 android:onClick="onClickAvecHandler"  
 />  
  
</RelativeLayout>

* Modifier le contenu de ActivityMain.java

**package** fr.eni.demonstrationm07d01;  
  
**import** android.os.AsyncTask;  
**import** android.os.Handler;  
**import** android.os.Message;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.ProgressBar;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 ProgressBar **pb** = **null**;  
 Button **btnAsyncTask** = **null**;  
 Button **btnHandler** = **null**;  
 MonHandler **handler** = **new** MonHandler();  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState)  
 {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **pb** = findViewById(R.id.***pb\_demonstration***);  
 **btnAsyncTask** = findViewById(R.id.***btn\_avec\_asynctask***);  
 **btnHandler** = findViewById(R.id.***btn\_avec\_handler***);  
 }  
  
  
 **public void** onClickGo(View view)  
 {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **for** (**int** i = 0;i<=10;i++)  
 {  
 **pb**.setProgress(i);  
 **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }).start();  
 }  
  
 **public void** onClickSansThread(View view)  
 {  
 **for** (**int** i = 0;i<=10;i++)  
 {  
 **pb**.setProgress(i);  
 **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 **public void** onClickAvecThread(View view)  
 {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **for** (**int** i = 0;i<=10;i++)  
 {  
 **pb**.setProgress(i);  
 **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }).start();  
 }  
  
 **public void** onClickAvecASyncTask(View view)  
 {  
 **new** Worker().execute();  
 }  
  
 **public void** onClickAvecHandler(View view)  
 {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 Message msgGo = **new** Message();  
 msgGo.**what** = 1;  
 **handler**.sendMessage(msgGo);  
  
 **for** (**int** i = 0;i<=10;i++)  
 {  
 Message msgEnCours = **new** Message();  
 msgEnCours.**what** = 2;  
 msgEnCours.**arg1** = i;  
 **handler**.sendMessage(msgEnCours);  
 **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 Message msgEnd = **new** Message();  
 msgEnd.**what** = 3;  
 **handler**.sendMessage(msgEnd);  
  
  
 }  
 }).start();  
 }  
  
  
 **class** Worker **extends** AsyncTask<Void,Integer,String>  
 {  
  
 @Override  
 **protected** String doInBackground(Void... voids) {  
 **for** (**int** i = 0;i<=10;i++)  
 {  
 publishProgress(i);  
 *//pb.setProgress(i);* **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 **return "fin"**;  
 }  
  
 @Override  
 **protected void** onPreExecute()  
 {  
 **super**.onPreExecute();  
 **btnAsyncTask**.setEnabled(**false**);  
 }  
  
 @Override  
 **protected void** onProgressUpdate(Integer... values)  
 {  
 **super**.onProgressUpdate(values);  
 **pb**.setProgress(values[0]);  
 }  
  
 @Override  
 **protected void** onPostExecute(String s)  
 {  
 **super**.onPostExecute(s);  
 **btnAsyncTask**.setEnabled(**true**);  
 }  
 }  
  
 **class** MonHandler **extends** Handler  
 {  
 @Override  
 **public void** handleMessage(Message msg)  
 {  
 **super**.handleMessage(msg);  
  
 **switch** (msg.**what**)  
 {  
 **case** 1: **btnHandler**.setEnabled(**false**); **break**;  
 **case** 2: **pb**.setProgress(msg.**arg1**); **break**;  
 **case** 3: **btnHandler**.setEnabled(**true**); **break**;  
 }  
 }  
 }  
  
  
}