**NETWORKING**

**Quando un’app ha bisogno di comunicare via Internet che cosa è necessario fare? Si faccia un esempio usando uno snippet di codice (senza dettagli ma solo con la struttura generale).**

Per eseguire operazioni di rete nell'applicazione, il manifest deve includere le seguenti autorizzazioni:

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

*<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />*

Per evitare di creare un'interfaccia utente che non risponde, non eseguire operazioni di rete sul thread dell'interfaccia utente. Per impostazione predefinita, Android 3.0 (livello API 11) e versioni successive richiede di eseguire operazioni di rete su un thread diverso dal thread dell'interfaccia utente principale; se non lo si fa, viene lanciata una ***NetworkOnMainThreadException***.

Il frammento *Activity* seguente utilizza un headless *Fragment* per incapsulare operazioni di rete asincrone.

**Il seguente abbozzo di classe NetworkTask mostrato viene utilizzato per ricevere da Internet dei dati nella forma di Stringa. Nella parte di ricezione (non dettagliata) viene eseguito un ciclo per 100 iterazioni e ad ogni iterazione viene ricevuta una stringa str mentre la stringa received\_data viene utilizzata per memorizzare tutti i dati ricevuti che alla fine verranno visualizzati nel TextView textResponse. Nel layout c’è una progressBar utilizzata per mostrare in quale iterazione ci troviamo. Questo codice non funziona. Perché? Come possiamo risolvere il problema?**

***class NetworkTask extends AsyncTask<Integer, Integer, String> {***

***@Override***

***protected void onPreExecute() {***

***progressBarDownload.setVisibility(ProgressBar.VISIBLE);***

***}***

***@Override***

***protected String doInBackground(Integer...values) {***

***//Do something***

***String received\_data = “”;***

***for (int i=0; i<100; i++) {***

***progressBarDownload.setProgress(i);***

***//Ricevi i dati nella stringa str***

***received\_data += str;***

***}***

***return received\_data;***

***}***

***@Override***

***protected void onPostExecute(String data) {***

***progressBarDownload.setVisibility(ProgressBar.INVISIBLE);***

***textResponse.setText(data);***

***}***

***}***

Il codice sopra scritto non funziona in quanto, quando il metodo doInBackground() è in esecuzione, non si può interagire con l’activity principale (dato che ci si trova in un thread separato), per cui non è possibile impostare il progresso in tale metodo (dovrebbe esser lanciato un errore). Per risolvere, in doInBackground() va eliminata la prima riga di codice all’interno del ciclo for, sostituendola con publishProgress(i): questo metodo invocherà onProgressUpdate(Integer i) sull’activity principale, per cui al suo interno andremo ad impostare il progresso della progressBar.

Inoltre, se il ciclo for passa ad i valori da 0 a 99, la progressBar non verrebbe mai riempita del tutto (non è un errore, ma è bene precisare). Infine, in onPostExecute() andrebbe settata la progress bar a 0 (non è un errore, ma è bene precisare).