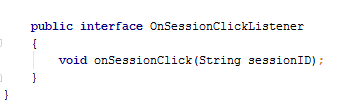
IL MECCANISMO DI CALLBACK

Come comunicano i fragment con le activity? Il meccanismo di callback ci permette di avvertire le activity che qualcosa è avvenuto sul fragment. Il funzionamento generale è che ogni activity deve implementare un’interfaccia, ed il fragment deve ottenere un riferimento all’interfaccia e chiamare il metodo dell’interfaccia implementato dall’activity.

Esempio: avvertire l’activity che c’è stata una pressione allungata su di una lista di un fragment

1.Creazione dell’interfaccia:

Conviene crearla nel Fragment visto che è legata al fragment più che all’activity. Apriamo il fragment e scriviamo in fondo:



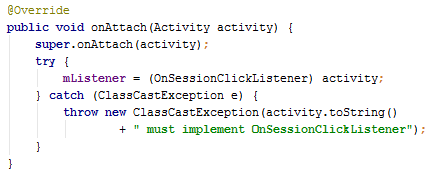
2. Oggetto interfaccia

Adesso ci serve un riferimento a questa interfaccia in modo da poterne chiamare i metodi, perciò creiamo all’inizio del fragment, insieme alle altre variabili



3. Collegamento dell’oggetto interfaccia

Questo variabile interfaccia non è collegata ancora a niente, dobbiamo collegarlo all’activity, possiamo farlo nel onAttach del fragment

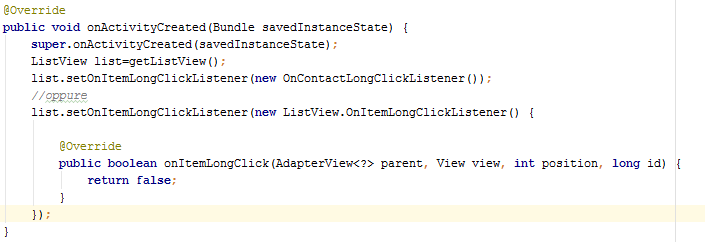


Questo metodo fa un controllo e lancia una eccezione se la classe non ha implementato il metodo.

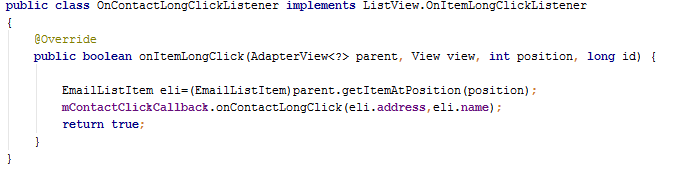
4. Rilevamento click lungo su lista

(nota mContactClickCallback sarebbe l’mListener delle foto precedenti, e il onContactLongClick() sarebbe onSessionClick dell’interfaccia precedente)

Creiamo una classe interna al fragment (o si può fare anche una classe anonima al momento) per mettere il longclicklistener sulla nostra lista :



E in entrambi i casi il metodo onItemLongClick conterrà

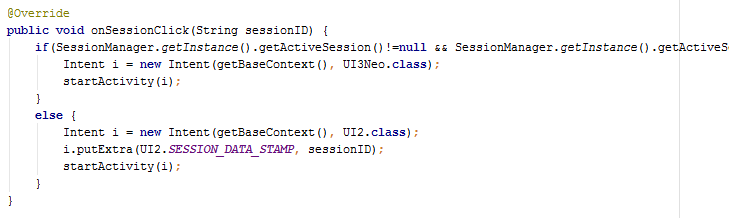


Il return true; serve a dire che il click lungo è stato gestito

5. Implementazione metodo su activity



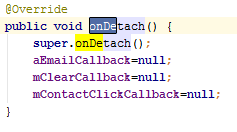
Aggiungiamo l’implement su TUTTE le activity che contengono o possono contenere il fragment



E implementiamone il metodo

6. Rimozione del callback

Per evitare che un frammento in memoria chiami il metodo di callback anche se non è più visibile sullo schermo annulliamo il callback quando il fragment non è più collegato all’activity



FINE

Ps nei fragment sono stati chiamati metodi in diversi punti del sul lifecycle perché il fragment in alcuni punti ha le cose pronte ed in altri no, in 99% dei casi se un metodo chiamato su un fragment da errore perché chiamato su un oggetto nullo, è perché quell’oggetto non è ancora stato inzializzato a quel punto del lifecycle, ad esempio il metodo getListView dei ListFragment può essere chiamato in qualunque momento, ma se viene chiamato ad esempio sull’onAttach ritorna nullo perché in quel momento la lista non è ancora stata inzializzata, per quel motivo è sul onAcitivityCreated()