Volley

La nostra applicazione è stata sviluppata con lo scopo di sfruttare al massimo le potenzialità messe a disposizione dal portale web marvel developer, pertanto l’interazione con questo, mediante l’uso di internet, è fondamentale per quasi ogni azione svolta dalla nostra app. E’ stato necessario gestire queste interazioni con la rete mediante l’utilizzo di thread, in modo da non appesantire il main thread e non causare di conseguenza blocchi dell’interfaccia grafica. Per eseguire queste operazioni, a livello implementativo, abbiamo optato per l’utilizzo di una ben nota libreria android che adempie a tali scopi: volley.

Volley infatti permette di gestire in modo molto semplice le richieste http/https, utilizzando per i suoi scopi l’uso di thread e di buffer con strutture a coda per gestire le diverse richieste (solitamente di tipo FIFO). Questa libreria inoltre fa utilizzo di strutture di cache per salvare i risultati delle ricerche e garantire maggior efficienza.

Per utilizzare volley occorre dapprima fare due cose: dichiarare il suo utilizzo nel file di gradle ed inserire nel manifest.xml i permessi dell’app relativi agli accessi ad internet.

Dopo di che nell’applicazione sono state definite 5 classi dedicate all’uso di volley, ed utilizzate per gestire richieste atte a reperire tutte le informazioni necessarie, relative rispettivamente a: eroi, fumetti, serie, eventi ed autori.

Queste classi sono state sviluppate seguendo la medesima struttura, e risultano pertanto classi “standardizzate”. La loro definizione infatti si apre con la dichiarazione di due attributi, uno relativo alla key utilizzata per effettuare le richieste, e l’altro relativo all’url di base a cui verranno inviate queste richieste. Successivamente sono stati dichiarati i metodi che, se invocati, permettono di formulare il corretto url (generalmente prendendo in input l’id di un oggetto di interesse) ,e di conseguenza effettuare tutti i tipi di richieste necessarie relative alla tipologia della volley in questione.

Questi metodi, una volta formulato il corretto url, convergono tutti nell’utilizzo di un metodo di api call, il quale inizializza la richiesta e la inserisce nella coda. (ricordiamo che tutte le richieste che formuliamo sono di tipo get).

Troviamo infine le due funzioni che gestiscono la risposta alla richiesta formulata. Nel caso in cui la risposta non sia stata correttamente ricevuta infatti entra in gioco un metodo di error response, il quale prima cerca di capire se l’errore è dipeso da un malfunzionamento della connessione ad internet (segnalandolo eventualmente all’utente mediante un toast), e poi scrive sul log un messaggio di errore.

Nel caso in cui invece la risposta arrivata si riveli corretta, entra in funzione il metodo di onResponse, che va a prendere in input il file di tipo json restituito nella risposta di get. A questo punto il metodo istanzia un oggetto di tipo GSON che, combinato con il meccanismo di reflection messo a disposizione da java, verrà utilizzato per effettuare la conversione di array json in array di oggetti java (opportunamente definiti nel model per il corretto mappaggio degli attributi).

Al termine dell’operazione, se la lista non è vuota, viene scritto nel log un messaggio con il numero degli oggetti inseriti nell’array. Invece, in caso di lista vuota, si andrà a segnalare all’utente mediante l’uso di un toast che non stati trovati risultati.

Infine viene fatta un’ operazione di callback al metodo di fill appartenente all’oggetto che ha istanziato la volley e che ne ha invocato le operazioni. Tale metodo utilizzerà i dati scaricati mediante la volley per popolare le recycler view presenti nella user interface.

La classe volley si chiude così con un catch che cattura eventuali eccezioni legate all’uso del json.

n.b. la nostra scelta di utilizzare volley è anche dipesa dal fatto che ci siamo ritrovati ad operare con file di piccole dimensioni, se invece avessimo dovuto gestire file di dimensione maggiore, questa non sarebbe stata la scelta più efficiente ,poiché questa libreria è ottimizzata per il download di file relativamente piccoli(<100mb). Pertanto, con file di dimensioni maggiori, sarebbe stato più corretto utilizzare il download manager.

Menù e pagina about us

Abbiamo deciso di inserire all’interno della nostra applicazione un’intera pagina dedicata al team che ha lavorato allo sviluppo della stessa. Si tratta di una pagina molto completa, che presenta dapprima il logo della start up e una breve descrizione del team di sviluppo, e che successivamente riporta in forma di bottoni e voci molto semplici ed esplicative tutti i link e le pagine che permettono di reperire tutte le informazioni a noi relative disponibili nell’intera rete internet. L’utente può ad esempio trovare le nostre pagine facebook ,twitter e instagram, con le quali è in grado di mettersi in contatto con noi e seguire in tempo reale le nostre attività, ma anche la nostra pagina github, che gli permette di accedere a tutto quel codice che abbiamo deciso di lasciare pubblico. È presente inoltre la possibilità di mettersi in contatto con il nostro team a scopo collaborativo-professionale, tramite la nostra mail certificata. Non abbiamo poi tralasciato nemmeno la pagina relativa alla nostra applicazione sul play store, che permette di accedere velocemente a tutte le caratteristiche e recensioni relative alla nostra app, oppure la nostra pagina youTube, riportante video relativi ai lavori svolti presso la nostra start up. Infine non può mancare una voce per accedere al nostro sito web, la fonte con le informazioni più complete e dettagliate sul nostro team e la nostra impresa.

Parlando invece degli aspetti implementativi, abbiamo deciso di inserire la nostra pagina about us come unica voce di un classico menù android, posizionato come consuetudine in alto a destra nell’applicazione ,e recante l’immagine dei tipici 3 pallini disposti in verticale. Vogliamo precisare che la nostra pagina about us non è stata realizzata mediante l’utilizzo di un file xml, ma invece attraverso una libreria java reperita online, che permette di generare la pagina e tutto il suo contenuto in modo dinamico basandosi unicamente sul codice scritto in java.

Il funzionamento della libreria è molto semplice, tutte le caratteristiche della pagina sono definite e gestite da un about us activity in java. Questa infatti ci permette di associare tutte le voci e i relativi bottoni ai link delle corrispondenti pagine web, di definire la descrizione del nostro gruppo, e di scegliere le immagini da mostrare. Tutti i link e le stringhe sono stati definiti nello strings.xml, compresa la descrizione, che è stata localizzata nelle 3 lingue riportate nella corrispondente sezione di questa relazione.

Tornando invece al menù, questo è stato definito attraverso la procedura classica, ovvero mediante la creazione di un file menu.xml presente in una sottocartella menu della directory res. Lo stile è anch’esso molto semplice, si tratta di una classica piccola sezione che compare al click del pulsante di menu.(Poiché per noi le scelte più semplici, spesso si rivelano anche le più efficaci). Il menu.xml poi è stato collegato all’apposito bottone di menu mediante il suo id. La semplicità di questo menu non ha reso necessario utilizzare le clausole e gli attributi che android rende disponibili per tali situazioni.

Successivamente il menu è stato collegato alla main activity definendo nella stessa un metodo onCreateOptionsMenu, che non fa altro che “gonfiare” il menù , rendendo l’icona visibile nella main activity. Infine abbiamo implementato il click sul menu stesso, andando a definire sempre nella main activity il metodo onOptionsItemSelected, che intercetta il click dell’ utente, e fa partire la relativa activity di about us.

n.d. Non essendo il nostro gruppo un vero team di sviluppo, e non avendo di certo una start up, abbiamo deciso di non creare realmente tutte le pagine web relative a Tiburtina valley inc. e presentate nella nostra about page. Tuttavia tutte le voci sono state effetivamente collegate a link e tutti i bottoni implementati, utilizzando come pagine di riferimento (youtube,facebook, twitter o instagram che siano) pagine scelte casualmente cercando il nome “tiburtina valley” nei vari siti web. L’unica pagina realmente relativa a noi è quella di github. (inoltre il link al play store porta ad una pagina vuota, non avendo noi effettuato il deploy della nostra applicazione su di esso).