

WORLD BANK

Progetto Mobile Programming 2018

Gruppo - Clashers

Mauro Ficarella, Valentina Sisti, Martina Turbessi

Indice

1. WorldBank
2. Funzionalità dell'app
3. Struttura, design app e layout
4. Colori & Font
5. Lingua
6. Requisiti, AndroidManifest.xml
7. Database
8. Volley
9. Gestione della connessione

1 – World Bank

La WorldBank è un'istituzione internazionale che mette a disposizione tramite il proprio sito (<http://www.worldbank.org/>) un servizio di acquisizione di dati storici riguardanti diverse variabili, tra cui:

- 16695 indicatori
- 21 argomenti
- 43 sorgenti
- 304 paesi (compresi anche gli aggregati, ad es. “Europa”)

Queste informazioni sono reperibili in formato json sul sito (<http://data.worldbank.org/>).

2 – Funzionalità dell'applicazione

L'applicazione permetterà all'utente di poter effettuare una ricerca per:

- Paese —> Argomento —> Indicatore
- Argomento —> Indicatore —> Paese

Per ogni variabile verrà visualizzata una finestra di dialogo che mostra nel caso di Argomento e Indicatore la loro descrizione, e nel caso di Paese la capitale.

Finita la scelta, l'applicazione mostrerà un grafico a linee (LineChart) che eventualmente potrà essere salvato dall'utente su richiesta.

Nella schermata iniziale inoltre l'utente potrà decidere se visualizzare:

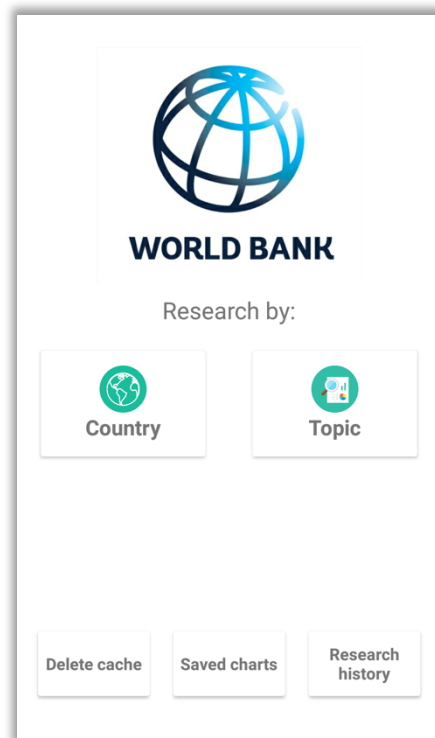
- Tutte le immagini salvate;
- Tutte le ricerche effettuate in precedenza tramite le quali potrà anche risalire al grafico collegato alla ricerca stessa;
- Pulire la cache.

3 – Struttura, design app e layout

Per poter implementare tutte queste funzionalità abbiamo suddiviso la struttura dell'applicazione in 7 activity:

- **activity_main:** Main activity dalla quale è possibile accedere alle 2 modalità di ricerca, ai grafici salvati, alle vecchie ricerche effettuate e eventualmente cancellare la cache;
- **activity_topic:** Activity che contiene una lista di tutti gli argomenti della WorldBank, dei quali l'utente potrà sceglierne uno. Per ogni argomento della lista selezionato verrà visualizzato un dialog contenente la sua descrizione;
- **activity_indicator:** Activity che contiene una lista di una porzione di indicatori della WorldBank, in base alla scelta precedente dell'argomento, dei quali l'utente potrà sceglierne uno. Per ogni indicatore della lista selezionato verrà visualizzato un dialog con la sua descrizione;
- **activity_country:** Activity che contiene una lista di tutti i paesi/aggregatori della WorldBank dei quali l'utente potrà sceglierne uno. Per ogni paese/aggregatore della lista selezionato verrà visualizzato un dialog che mostrerà la capitale, la latitudine e la longitudine di quel paese/aggregatore;
- **activity_chart:** Activity che, in base alle scelte effettuate dall'utente, mostrerà un grafico a linee (LineChart);
- **activity_saved_charts:** Activity dove è possibile visualizzare tutte le immagini, relative ai grafici, salvate dall'utente;
- **activity_full_image:** Activity che mostra l'immagine contenente un singolo grafico salvato la quale eventualmente potrà essere condivisa su diverse applicazioni;
- **activity_offline_data:** Activity che mostra la cronologia delle ricerche effettuate.

3.1 – activity_main



La home page dell'applicazione è costituita da dei linear layout e da delle card view.

Ognuna di queste card view funge da bottone il quale reindirizza dalla home ad un'activity differente in base a cosa seleziona l'utente:

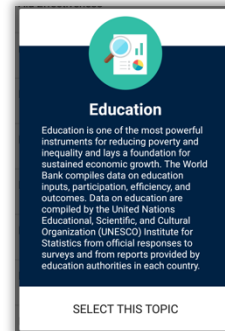
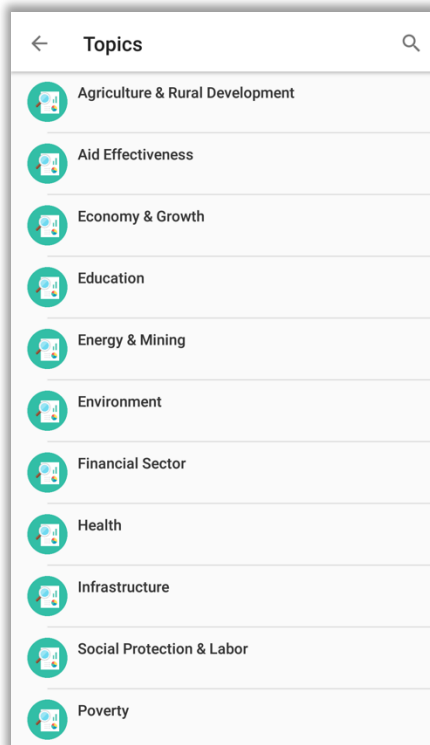
- “Country” reindirizza l'utente ad activity_country;
- “Topic” reindirizza l'utente ad activity_topic;
- “Research history” reindirizza l'utente ad activity_offline_data;
- “Saved charts” reindirizza l'utente ad activity_full_image.

Inoltre abbiamo una card view “Delete cache” che permetterà all'utente di cancellare la cache dell'applicazione (cancella tutti i json e le requests salvate).

La visualizzazione di questa activity è bloccata in modalità portrait.

Per questa activity sono state create versioni adatte alle risoluzioni più comuni dei vari smartphone presenti all'interno del database in AndroidStudio.

3.2 – activity_topic



Si arriva a questa activity in due modi, proveniando da:

- activity_main se l'utente ha cliccato sulla CardView "Topic";
- activity_country se l'utente ha selezionato un paese/aggregatore.

Quest'activity mostrerà una lista di argomenti i cui nomi saranno richiesti e selezionati con una query all'URL collegato al file json

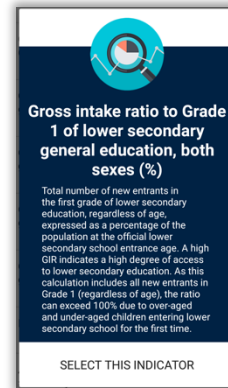
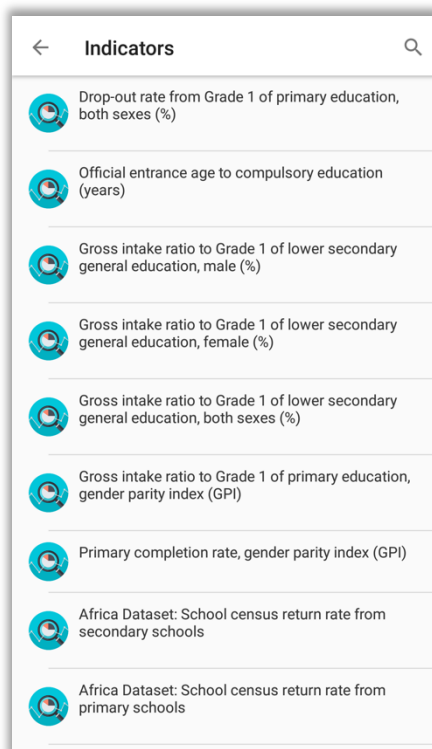
(<http://api.worldbank.org/v2/topics/?format=json>).

In quest'activity l'utente potrà ricercare manualmente o tramite una barra di ricerca un argomento nella lista (la ricerca parte dal terzo carattere inserito). Cliccando su un elemento della lista si aprirà un dialog nel quale si potrà leggere la descrizione dell'argomento ed eventualmente decidere se ritornare alla lista degli argomenti o selezionare quell'argomento tramite il bottone "Select this topic".

Una volta selezionato un argomento l'applicazione dovrà memorizzare l'ID dell'argomento selezionato che verrà usato per caricare activity_indicator con la giusta lista di indicatori inerenti all'argomento selezionato tramite una query. Inoltre l'utente potrà tornare alla schermata precedente cliccando sulla freccia in alto a sinistra nella toolbar.

Anche questa activity è visibile solo in modalità portrait.

3.3 – activity_indicator



Verremo reindirizzati in questa activity dall'activity_topic. Quest'activity mostrerà una lista degli indicatori i cui nomi saranno richiesti e selezionati con una query all'URL collegato al file `indicators.json`

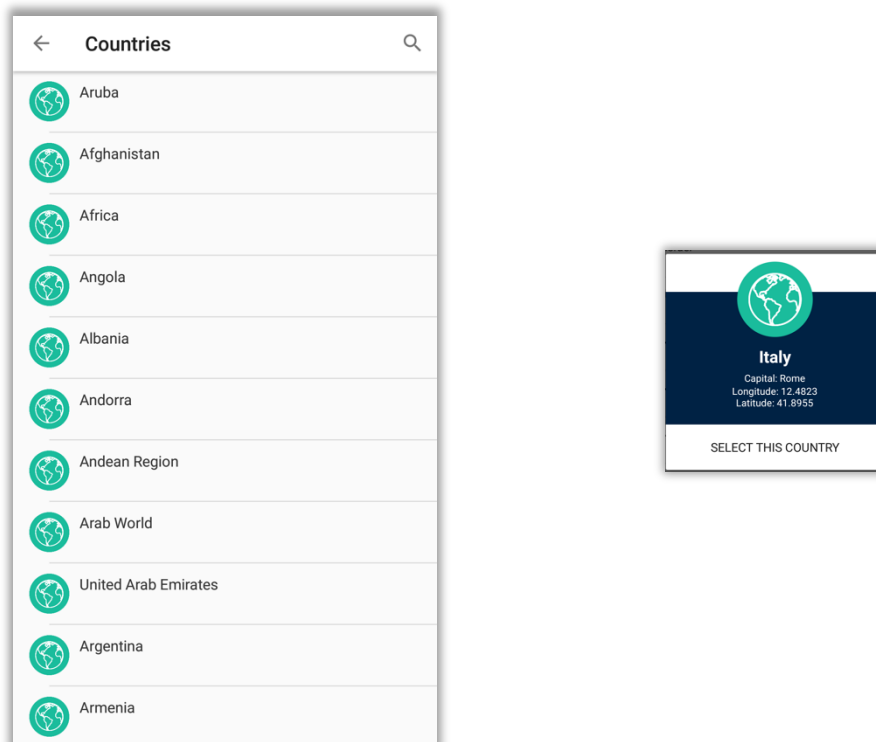
([http://api.worldbank.org/v2/topics/"+fullQuery.getTopic\(\).getId\(\)+"/indicators/?per_page=3500&format=json](http://api.worldbank.org/v2/topics/)).

In quest'activity l'utente potrà ricercare manualmente o tramite una barra di ricerca un indicatore nella lista (la ricerca parte dal terzo carattere inserito). Cliccando su un elemento della lista si aprirà un dialog nel quale si potrà leggere la descrizione dell'indicatore ed eventualmente decidere se ritornare alla lista degli indicatori o selezionare quell'indicatore tramite il bottone “Select this indicator”. Una volta selezionato un indicatore l'applicazione dovrà memorizzare l'ID dell'indicatore selezionato che verrà usato nella query per la richiesta del grafico finale.

Inoltre l'utente potrà tornare alla schermata precedente cliccando sulla freccia in alto a sinistra nella toolbar.

Anche questa activity è visibile solo in modalità portrait.

3.4 – activity_countries



Si arriva a questa activity in due modi, proveniendo da:

- activity_main se l'utente ha cliccato sulla CardView "Country";
- activity_indicator se l'utente ha selezionato un indicatore.

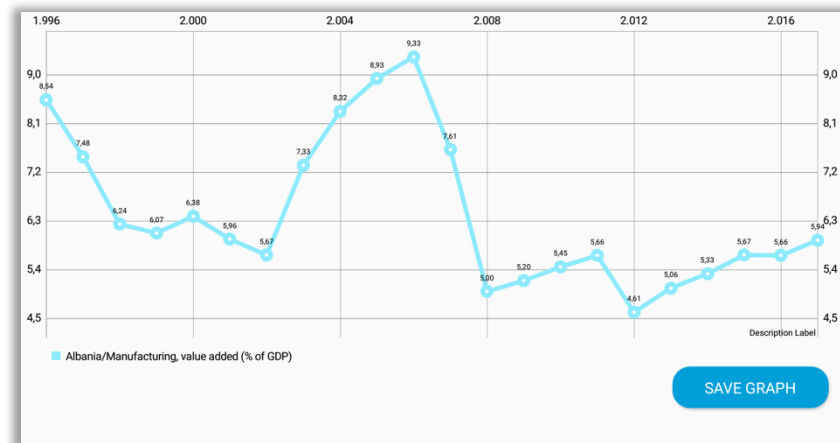
Quest'activity mostrerà una lista di paesi i cui nomi saranno richiesti e selezionati con una query all'URL collegato al file json (http://api.worldbank.org/v2/countries/?per_page=304&format=json).

In quest'activity l'utente potrà ricercare manualmente o tramite una barra di ricerca un paese nella lista (la ricerca parte dal terzo carattere inserito). Cliccando su un elemento della lista si aprirà un dialog nel quale si potrà leggere la capitale, la longitudine e la latitudine del paese (se si tratta di aggregatori queste non sono previste) ed eventualmente decidere se ritornare alla lista dei paesi o selezionare quel paese tramite il bottone "Select this country". Una volta selezionato un paese l'applicazione dovrà memorizzare l'"ISO2CODE" che verrà usato nella query per la richiesta del grafico finale.

Inoltre l'utente potrà tornare alla schermata precedente cliccando sulla freccia in alto a sinistra nella toolbar.

Anche questa activity è visibile solo in modalità portrait.

3.5 – activity_chart



Si arriva a questa activity in tre modi, proveniendo da:

- activity_country se l'utente è arrivato dalla sequenza di ricerca Argomento —> Indicatore —> Paese;
- activity_indicator se l'utente è arrivato dalla sequenza di ricerca Paese —> Argomento —> Indicatore;
- activity_offline_data.

In quest'activity l'utente potrà visualizzare il grafico a linee (LineChart) della ricerca effettuata ed eventualmente salvarlo tramite il bottone "Save graph".

I dati del grafico verranno richiesti e selezionati con una query all'URL collegato al file json ([http://api.worldbank.org/v2/countries/"+fullQuery.getCountry\(\).getIso2Code\(\)+"/indicators/"+fullQuery.getIndicator\(\).getId\(\)+"/?per_page=3500&format=json](http://api.worldbank.org/v2/countries/)) contenente anche i dati riguardanti la country e l'indicator selezionati precedentemente.

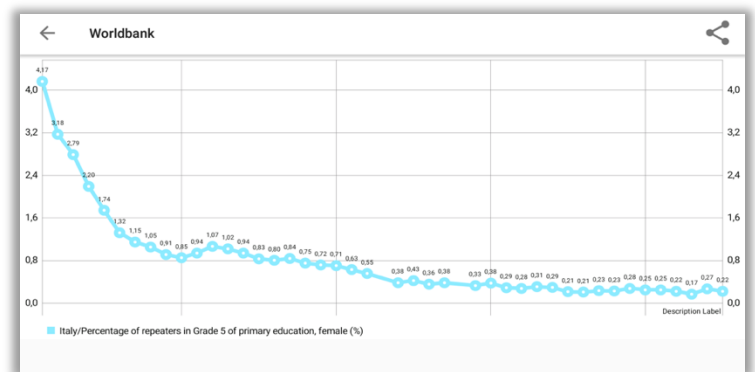
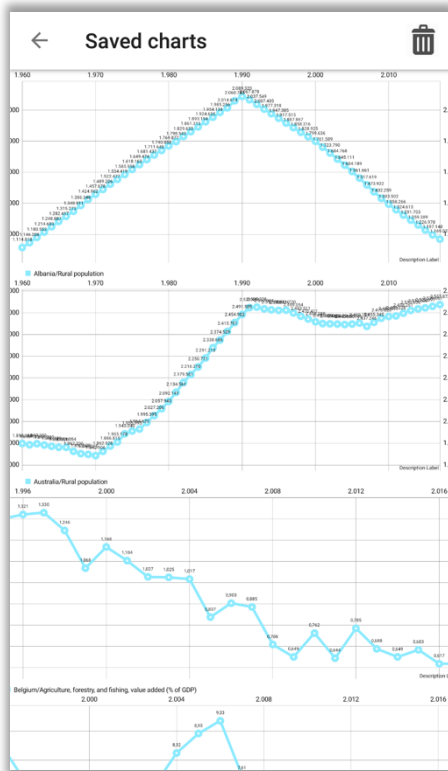
Inoltre in basso a sinistra verranno indicati il paese e l'indicatore della ricerca.

La visualizzazione di questa activity è bloccata in modalità landscape.

Per questa activity sono state create versioni adatte alle risoluzioni più comuni dei vari smartphone presenti all'interno del database in AndroidStudio.

Inoltre in questa activity, al momento del salvataggio del grafico, verranno richiesti a runtime i permessi per leggere e scrivere sulla memoria esterna (**READ_EXTERNAL_STORAGE/WRITE_EXTERNAL_STORAGE**).

3.6 – activity_saved_charts & activity_full_image

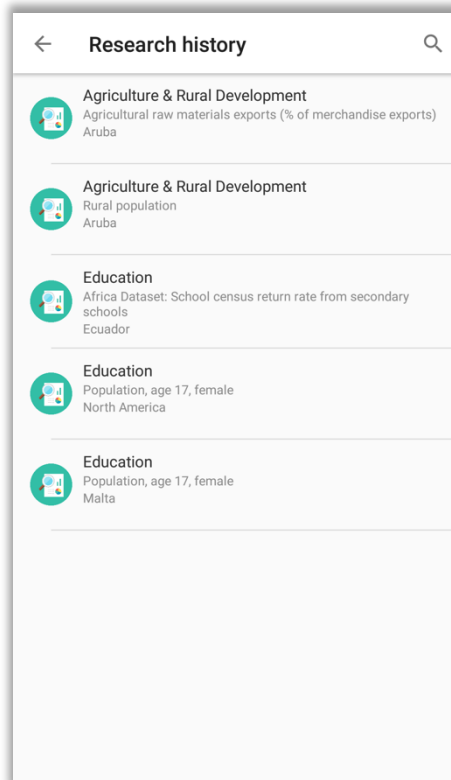


Verremo reindirizzati in activity_saved_charts cliccando sulla card “Saved charts” nella activity_main.

In quest’activity l’utente potrà visualizzare le immagini dei grafici salvate in precedenza e scegliere se aprirne una o tornare indietro. In alto a destra nella toolbar c’è un’icona raffigurante un cestino tramite la quale si può cancellare l’intera lista di immagini salvate.

L’immagine aperta verrà visualizzata in modalità landscape tramite l’activity_full_image e l’utente potrà decidere se condividerla tramite app esterne.

3.7 – activity_offline_data



Verremo reindirizzati in `activity_offline_data` cliccando sulla card “Research history” nella `activity_main`. In quest’activity verrà visualizzata una lista delle ricerche già effettuate e l’utente potrà ricercarne una manualmente o tramite una barra di ricerca (la ricerca parte dal terzo carattere inserito).

Cliccando su un elemento della lista arriveremo in `activity_chart` dove verrà mostrato il grafico corrispondente ricerca selezionata.

Inoltre l’utente potrà tornare alla schermata precedente cliccando sulla freccia in alto a sinistra nella toolbar.

4 – Colori & Font

Per i colori e il font dell'applicazione abbiamo consultato il documento messo a disposizione dalla WorldBank chiamato “Branding and Visual Identity Guidelines”.

4.1 – Colori:



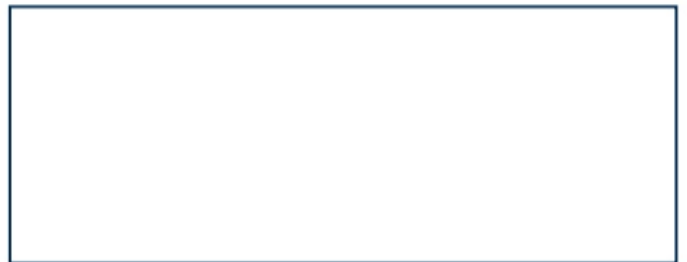
Solid Blue: HEX #002244



Bright Blue: HEX #009FDA



Black: HEX #000000



White: HEX #FFFFFF

4.2 – Font:

Il font primario della WorldBank è “Andes” il quale però è a pagamento. Per questo motivo abbiamo deciso di utilizzare il font secondario “Arial”. Di questo font abbiamo usato le due versioni Regular e Bold.

5 – Lingua

La lingua di default è la lingua inglese.
Non sono previste lingue secondarie.

6 – Requisiti, AndroidManifest.xml

Quest'applicazione è installabile solo su dispositivi sui quali è installato almeno Android 6 (API level 23).

Inoltre l'applicazione dovrà richiedere le seguenti autorizzazioni:

- Accesso a internet: serve perché, per effettuare le ricerche, l'applicazione utilizzerà i file json messi a disposizione dal sito della WorldBank, tramite un URL;
- Accesso all'archivio: serve per poter salvare/leggere nell'archivio esterno le immagini riguardanti i grafici richiesti dall'utente;
- Accesso allo stato della connessione: serve per poter controllare se è attiva o meno la connessione a internet.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

7 – Database

Il database si occupa di memorizzare i file json in base alle ricerche effettuate dall'utente. Questo database è formato da due tabelle:

- "json_url" : tabella dove verranno salvati gli URL e i corrispettivi file json relativi alle query quando l'utente effettua le ricerche;
- "requests" : tabella dove verranno salvate le ricerche effettuate in modo da poter essere riutilizzate.

La classe DBHelper estende la classe SQLiteOpenHelper che si occupa delle seguenti cose:

- Apertura del database;
- Creazione delle tabelle.

Per estendere tale classe è obbligatorio implementare i metodi *onCreate()* e *onUpgrade()/onDowngrade()*. Ad esempio la *onCreate()* usa il linguaggio SQL per gestire le tabelle nel seguente modo:

```
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String query = "CREATE TABLE IF NOT EXISTS "+TABLE_URL_NAME+" (\n" +
        "url TEXT NOT NULL PRIMARY KEY,\n" +
        "json TEXT\n" +
        ");";
}
```

```

String query1 = "CREATE TABLE IF NOT EXISTS "+TABLE_REQUEST_NAME+" (\n" +
    "json TEXT,\n" +
    "topic TEXT,\n" +
    "indicator TEXT,\n" +
    "country TEXT,\n" +
    "PRIMARY KEY (topic, indicator, country)\n" +
    ");";

sqliteDatabase.execSQL(query);
sqliteDatabase.execSQL(query1);
}

```

Al fine di poter inserire dati nel database viene creato un `SQLiteDatabase`.

Gli elementi verranno inseriti all'interno delle tabelle nel seguente modo:

```

public void addURL(String url, String json) {
    ContentValues values = new ContentValues();
    values.put("url", url);
    values.put("json", json);
    db.insert(TABLE_URL_NAME, null,
        values);
}

public void saveRequestIntoDatabase(String json, String topic, String indicator, String country) {
    ContentValues values = new ContentValues();
    values.put("json", json);
    values.put("topic", topic);
    values.put("indicator", indicator);
    values.put("country", country);
    db.insert(TABLE_REQUEST_NAME, null,
        values);
}

```

Inoltre, tramite cursore, si può ottenere una request salvata in precedenza.

Infine sono presenti due metodi, `deleteAllJson()` e `deleteSavedRequests()`, che si occupano, rispettivamente, di cancellare tutti i json dalla tabella contenente gli url "TABLE_URL_NAME" e di cancellare tutti i dati riguardanti i json salvati nelle requests nella tabella "TABLE_REQUEST_NAME".

8 – Volley

L'accesso alla rete è un'attività molto comune delle app Android e che è necessariamente condizionata da tempi di latenza più o meno significativi. Tale operazione deve perciò essere svolta in maniera asincrona, su un thread secondario. Vista l'importanza di tale tipo di operazioni e le necessità di svolgerle correttamente, è stata predisposta una libreria chiamata Volley, che cura tutti gli aspetti di accesso alla rete:

- Gestione autonoma delle richieste e connessioni multiple;
- *Caching* delle risposte sia in memoria che su disco;
- Varie classi per il supporto dei tipi più comuni di richieste;
- Gestione delle priorità;
- Strumenti di log, debug e tracciamento delle attività.

Per usare Volley dobbiamo innanzitutto impostare la permission `INTERNET` all'interno del file *AndroidManifest.xml*.

In particolare nella nostra applicazione il Volley è usato per richiedere i file json resi disponibili attraverso le API della Worldbank.

Lavorare con Volley significa, prima di tutto, confrontarsi con due concetti fondamentali: Request e RequestQueue. Il protocollo HTTP è basato sull'interazione tra client e server in termini di richiesta-risposta: con un derivato della classe Request formuleremo la richiesta del file json e l'accoderemo nella RequestQueue.

In pratica non eseguiremo mai la richiesta immediatamente, ma ci limiteremo ad inserirla in una coda (la RequestQueue, appunto). Sarà poi Volley ad eseguirla appena possibile, secondo le sue politiche e le condizioni del sistema.

In ogni oggetto Request, saranno inclusi i riferimenti a due *listener*: uno derivante da *Response.Listener* invocato nel caso in cui la richiesta venga svolta con successo; l'altro, *Error.Listener*, entra in causa in caso di errori. In quest'ultimo listener utilizzeremo un oggetto *VolleyError* che non è altro che un'eccezione che maschera un errore HTTP.

Volley prevede già alcuni tipi di richiesta per le forme più comuni di comunicazione. In particolare noi usiamo nell'applicazione la *JSONArrayRequest* per prelevare l'intero contenuto remoto richiesto.

9 - Gestione della connessione

All'interno dell'applicazione è implementato anche un meccanismo di gestione della connessione. Infatti, ogni qualvolta l'utente cercherà di accedere a qualche funzionalità dell'applicazione che richiede la connessione alla rete, con quest'ultima che è disabilitata, verrà notificato tramite un `AlertDialog`. Questo `AlertDialog`, oltre ad informare l'utente della necessità della connessione ad internet per accedere ad una determinata funzionalità, permetterà, attraverso due bottoni, di essere reindirizzato alle impostazioni per abilitare rispettivamente il Wi-Fi e i Dati Mobili. In questo `AlertDialog` è presente inoltre un terzo bottone che darà all'utente la possibilità di uscire dall'applicazione qualora lo desideri. In alternativa si può tornare indietro ed accedere alle funzionalità offline dell'applicazione.