

# Meteo App

Android Studio

Zanichelli Mattia

SUPSI DTI – Sviluppo di applicazioni mobile DTI.M02074

Docente: Galli Vanni

Anno accademico 2021/22

Aprile 2022

## Indice

1. Introduzione .....	3
2. Struttura .....	3
2.1 Activities .....	3
2.1.1 MainActivity .....	3
2.1.2 DetailActivity .....	3
2.2 Database .....	3
2.2.1 DBSchema .....	3
2.2.2 DBHelper .....	3
2.2.3 LocationCursorWrapper .....	3
2.3 Fragments .....	3
2.3.1 ListFragment .....	3
2.3.2 DetailLocationFragment .....	4
2.4 Model .....	4
2.4.1 Location .....	4
2.4.2 LocationsHolder .....	4
2.5 Service .....	4
2.5.1 Fetcher .....	4
2.6 Utility .....	4
2.6.1 VolleyCallback .....	4
3. Conclusioni .....	5

# 1. Introduzione

Questo documento contiene una breve descrizione di come è stata sviluppata l'applicazione Android per la prima versione del progetto del corso, sviluppato usando Android Studio.

Per ogni componente del progetto viene fatta una breve spiegazione di cosa contiene e quali sono le sue funzionalità.

## 2. Struttura

Di seguito verranno elencati i packages del progetto e le classi che contengono.

### 2.1 Activities

Package contenente le Activity dell'applicazione.

#### 2.1.1 MainActivity

Questa è la Main Activity dell'applicazione, al suo interno vengono istanziati il database e il fragment contenente la lista delle locations dell'applicazione.

All'interno della classe vengono anche effettuate le operazioni per gestire le notifiche, attraverso il meccanismo di Worker.

#### 2.1.2 DetailActivity

Questa Activity è incaricata di mostrare i dettagli di una location una volta che l'utente seleziona un luogo elencato nella lista principale di locations. Viene semplicemente richiesta la location che l'utente ha selezionato nell'Activity precedente e viene caricata in modo da poter mostrare in seguito i suoi dettagli.

### 2.2 Database

Package contenente le classi che servono per la parte di database dell'applicazione.

#### 2.2.1 DBSchema

Questa classe serve principalmente per definire la struttura del nostro database. Comprende il nome della tabella che contiene i dati dell'applicazione, e le colonne: id e nome delle locations.

#### 2.2.2 DBHelper

Questa classe serve per inizializzare il database per l'applicazione, nel metodo "onCreate" viene infatti specificato il comando SQL per creare la tabella che conterrà le location inserite dall'utente.

#### 2.2.3 LocationCursorWrapper

Questa classe viene utilizzata per ritornare un oggetto di tipo Location che serve nel momento di lettura dei dati dal database. Si tratta di un Cursor che funge da "wrapper" per un oggetto Location.

### 2.3 Fragments

Questo package contiene i diversi Fragment dell'applicazione.

#### 2.3.1 ListFragment

Questo Fragment serve a rappresentare la lista di location attualmente presenti e che sono state aggiunte dall'utente. Si tratta di una semplice lista che elenca i nomi delle città aggiunte dall'utente.

In questa classe è inoltre presente la logica per la richiesta dei permessi all'utente per accedere alla posizione, in questo modo è possibile stabilire dove si trova l'utente e mostrare la meteo del luogo in cui si trova, inizialmente elencata come "Current Location", a cui viene poi cambiato il nome in base alla città in cui si trova l'utente.

### 2.3.2 DetailLocationFragment

In questo Fragment sono contenute le informazioni e le istruzioni per mostrare i dettagli di una location una volta selezionata dall'utente. Qui è presente la logica per mostrare l'icona adatta in base alle condizioni meteo (sole, pioggia, nuvole, ecc.). Altre informazioni riguardanti una location sono: il nome, la temperatura, l'umidità e la descrizione (cielo limpido, nuvoloso, nebbioso, ecc.).

## 2.4 Model

Questo package contiene i modelli dell'applicazione, ossia la classe Location e LocationsHolder, che funge da "contenitore" delle varie locations.

### 2.4.1 Location

Questa classe descrivere l'oggetto Location. È una classica classe Java contenente campi, getters e setters.

### 2.4.2 LocationsHolder

Questa classe serve per rappresentare il contenitore delle varie locations aggiunte dall'utente. Contiene la logica per ottenere le locations, aggiornare i dati di una location, leggere i dati dal database e ottenere una Location dato il suo UUID.

## 2.5 Service

In questo package è contenuta la classe che serve per ottenere le informazioni sulle location. Attraverso delle chiamate http vengono aggiornati i dati di tutte le locations presenti nella lista di locations aggiunte dall'utente.

### 2.5.1 Fetcher

L'unica classe contenuta in questo package serve per ottenere informazioni dal sito <https://api.openweathermap.org> tramite delle chiamate alle quali vengono ritornate le informazioni sottoforma di JSON. In particolare, ci sono due metodi: uno che ritorna un oggetto di tipo JSON data la longitudine e la latitudine di una location, il secondo dato il suo nome.

Questa classe estende la classe Worker e serve per le notifiche dell'applicazione, che purtroppo non sono riuscito a far funzionare.

Questa classe fa uso dell'API Volley che verrà spiegata in seguito.

## 2.6 Utility

L'ultimo package del progetto contiene un'interfaccia utile per fare le richieste http per ottenere le informazioni sulle locations.

### 2.6.1 VolleyCallback

Questa interfaccia serve principalmente per le richieste al sito Open Weather. È una API di Google (<https://google.github.io/volley>) che facilita le richieste http in formato JSON.

Per poter utilizzare le funzionalità di questa API la classe deve semplicemente implementare l'interfaccia e il suo metodo "onSuccess".

Dopo una breve ricerca online ho scelto di provare a utilizzare questa API e mi sono trovato abbastanza bene grazie alla sua facilità d'uso.

### 3. Conclusioni

L'applicazione funziona correttamente: dalla lista di location nel fragment principale è possibile aggiungere e rimuovere locations. Ci sono però alcuni dettagli che non sono ottimali: quando una location viene selezionata, l'icona che rappresenta la meteo non si aggiorna subito, bisogna uscire dai dettagli della location e rientrare per vedere effettivamente che l'icona mostrata è coerente con la descrizione della location.

Come accennato in precedenza, le funzioni di notifica non sembrano funzionare, nono stante il sistema accetti le notifiche da parte dell'applicazione.

In alcuni casi, ruotare il dispositivo causa dei crash dell'applicazione che non sono riuscito a risolvere.