

# PupilManager - Arbeiten mit Dateien

**PupilManager**  

-  Böhm Matthias
-  Eilmsteiner David
-  Garn Julian
-  Gubesch Christof
-  Harwöck Florian
-  Herzog Matthias
-  Hüttmaier Thomas
-  Kaindl Julian
-  Krenn Gabriel

**Edit Pupil**

Lastname  
Hüttmaier

Firstname  
Franz

City  
Vöcklamarkt

Is male



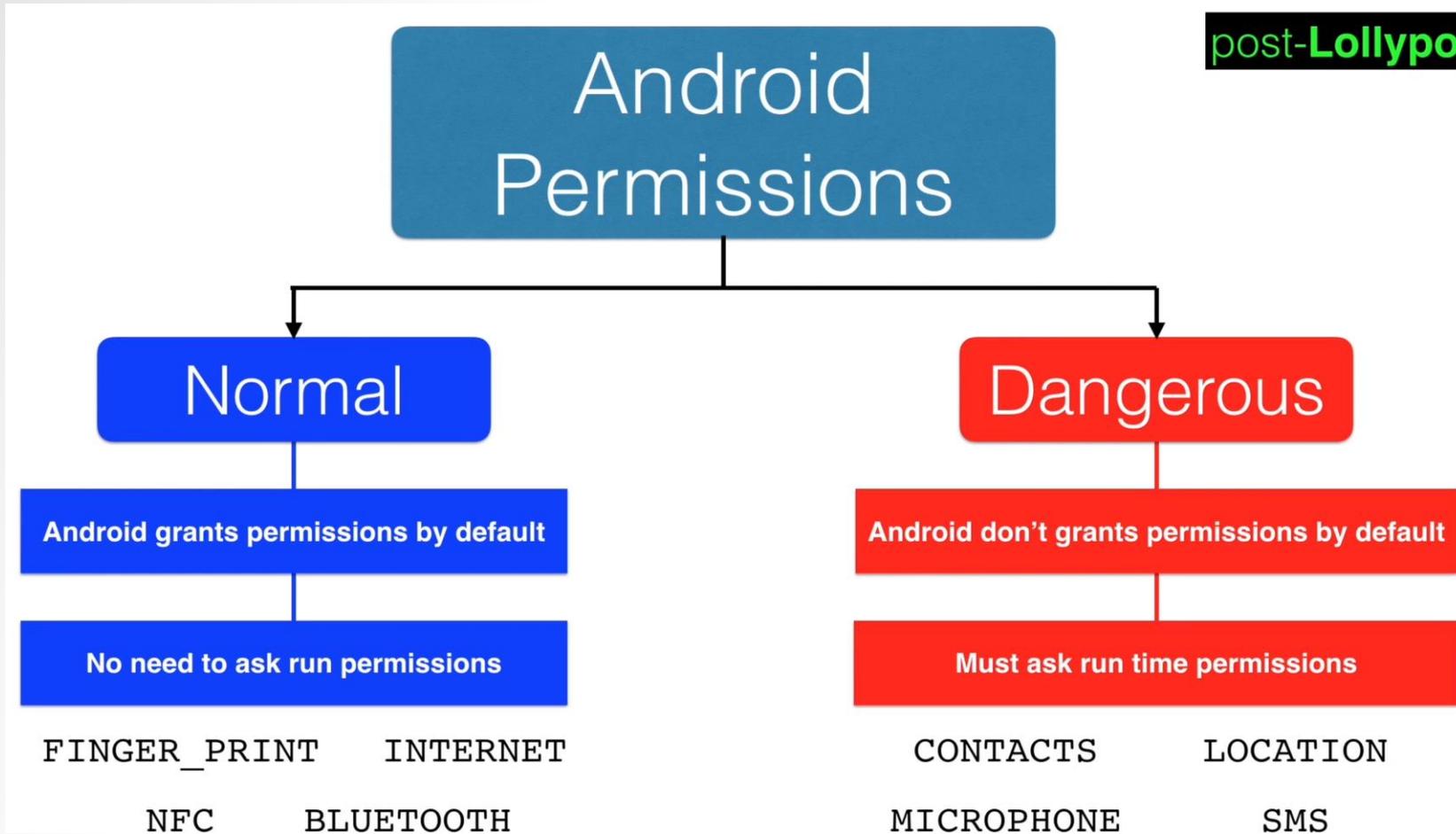
# Änderung bei Permissions seit Marshmallow

- Unterscheidung „Normal Permission“ und „DangerousPermission“
- Normale Permissions → wie bisher
- Gefährliche Permissions
  - Abfrage der Zustimmung des Benutzers zur Laufzeit

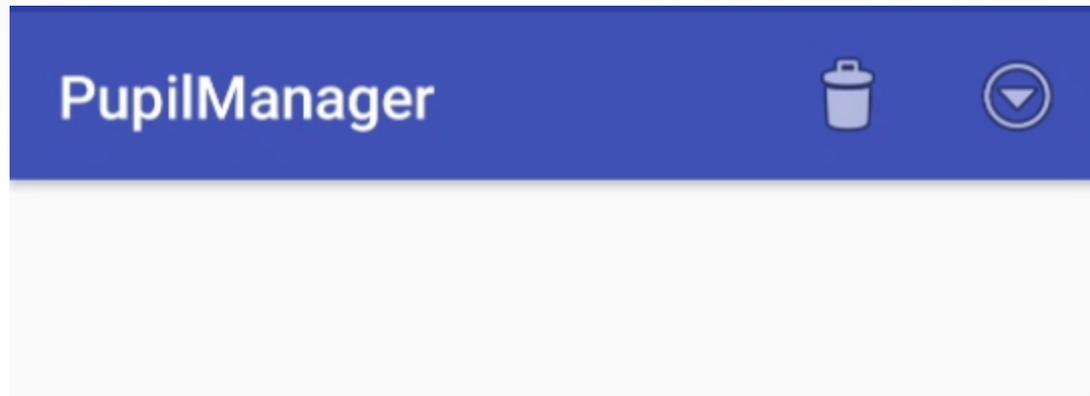
```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

# Permissions - Überblick

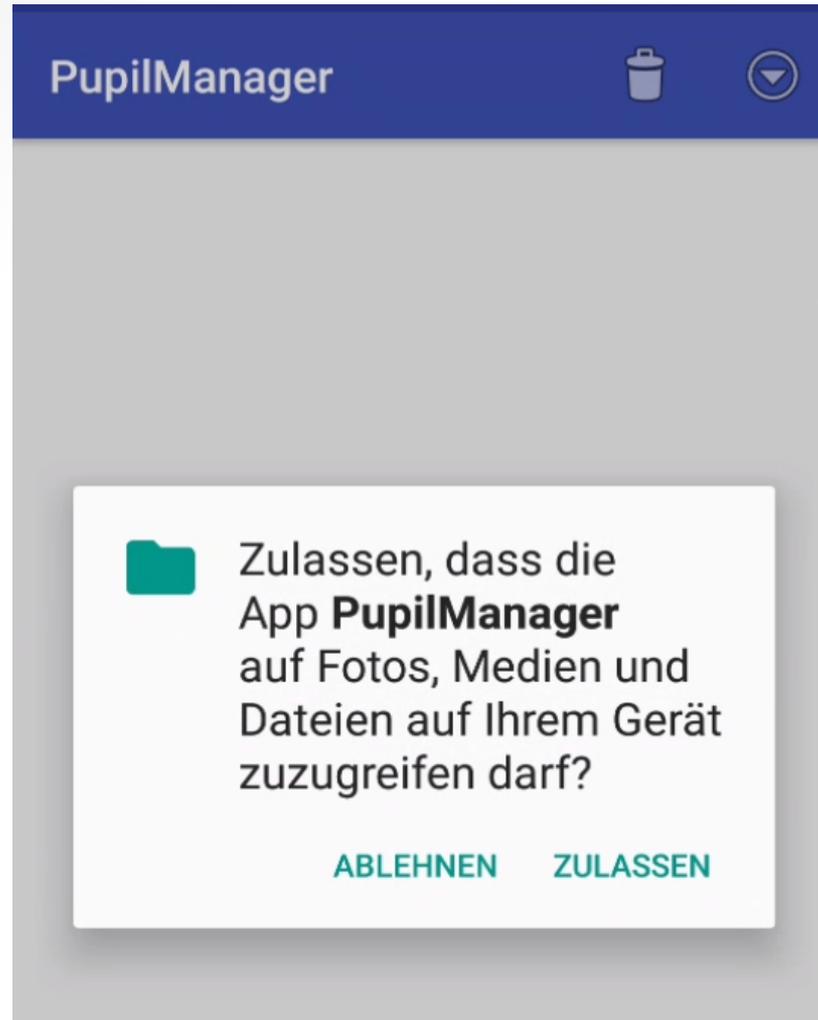
post-Lollipop



# Download-Button → File einlesen

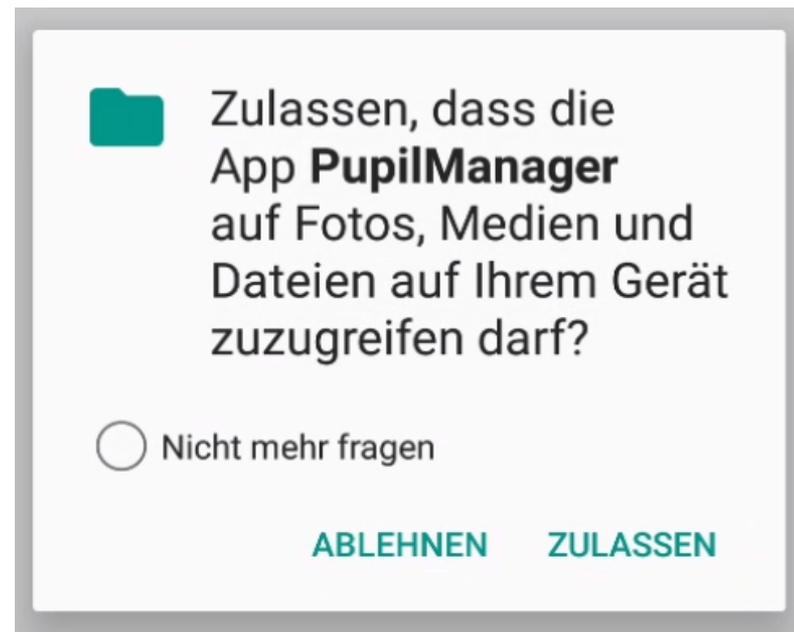
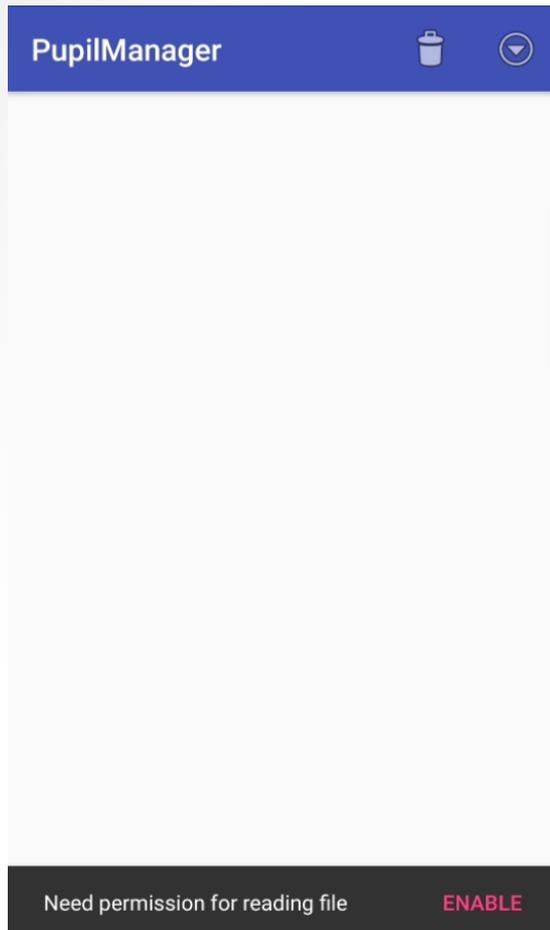


# Permission wird zur Runtime abgefragt



# Permission wurde abgelehnt

- Wieder Abfrage
  - Möglichkeit, endgültig abzulehnen



# Standardimplementierung

```
private fun loadPupilsFromSdCard() {
    if (ActivityCompat.checkSelfPermission( context: this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE) == PackageManager.PERMISSION_GRANTED) {
        Log.d(LOG_TAG, msg: "Permission is granted");
        val path :String = this.getExternalFilesDir(fileName).toString()
        val loadPupilsAsyncTask = LoadPupilsTask { it: List<Pupil>
            PupilDbTable(applicationContext).deleteAllPupils()
            it.forEach({ p -> PupilDbTable( context: this).add(p) })
            setTeachersAdapterToDb()
        }
        loadPupilsAsyncTask.execute(path)
    }else{
        Log.i(LOG_TAG, msg: "Permission is not granted");
        if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this,
            Manifest.permission.WRITE_EXTERNAL_STORAGE )) {
            // Permission wurde schon einmal abgefragt und abgelehnt
            Snackbar.make (findViewById(android.R.id.content), text: "Need permission for reading file", Snackbar.LENGTH_LONG
                .setAction( text: "ENABLE", { it: View! |
                    ActivityCompat.requestPermissions( activity: this,
                        arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE), PERMISSION_REQUEST_ID )
                })).show()
        }
        else{
            // Erstmalige Abfrage der Permission
            ActivityCompat.requestPermissions( activity: this,
                arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE), PERMISSION_REQUEST_ID )
        }
    }
}
```

# Vereinfachung durch Library Ask

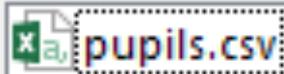
- <https://github.com/00ec454/Ask>

```
Ask.on( |Activity: this)
    .forPermissions(Manifest.permission.WRITE_EXTERNAL_STORAGE)
    .withRationales( ...rationalMessages: "WRITE_EXTERNAL_STORAGE permission need")
    .go()
```

# Datei auf Handy laden

- Im externen Speicher
  - `Environment.getExternalStorageDirectory()`
- Im Anwendungsverzeichnis

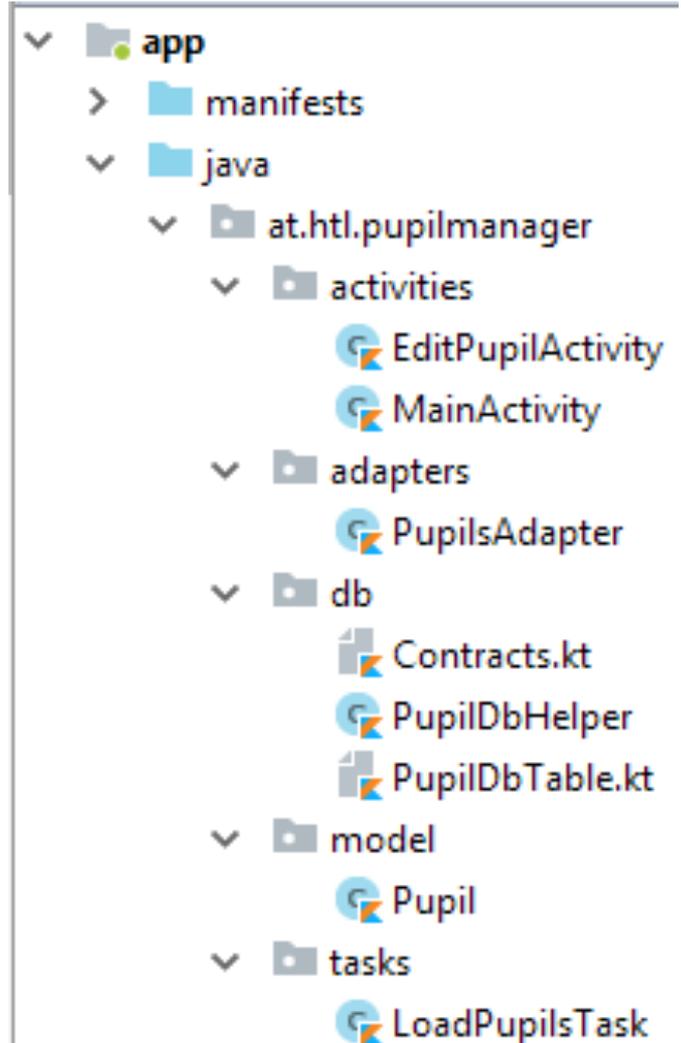
```
\Externer Speicher\Android\data\at.htl.pupilmanager\files\
```



- Oder im public Downloadverzeichnis
- Im Problemfall → Ressourcen
  - Raw-Verzeichnis anlegen
  - Mit App ausliefern → fixe Daten

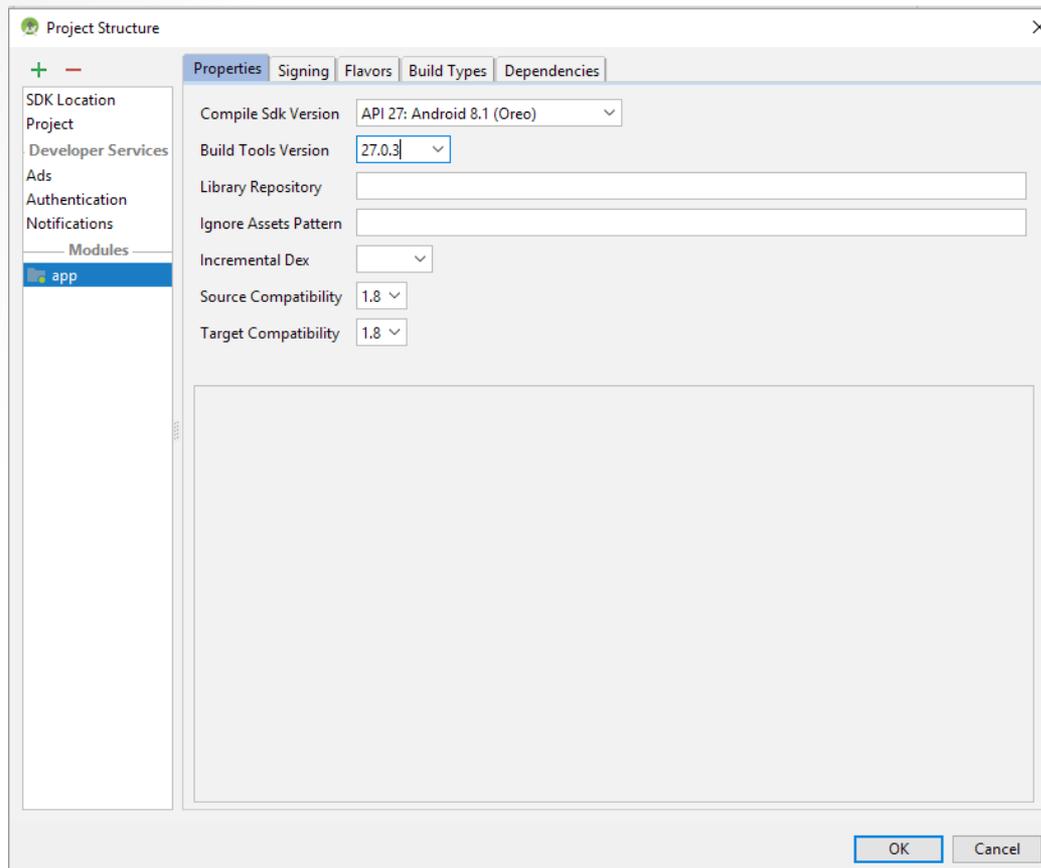
# Teilaufgaben

- Pupilkasse als Datenklasse anlegen
- Sqlite-DB für die Pupils anlegen
  - readAll(), add(), delete(), update()
- MainActivity mit RecyclerView
  - Image und FullName
  - Adapter mit ViewHolder
  - Einlesen der Daten im Hintergrund
- OnClickListener()
  - Aufruf der Bearbeitungsactivity
- EditPupilActivity
  - Bearbeiten der Schülerdaten



# Einlesen mit Files, Streaming-API

- Java 8 erforderlich
- <https://developer.android.com/studio/write/java8-support>



# Funktioniert nur auf neuesten Handies

```
val pupils : (Mutable)List<Pupil!>! = Files.lines(Paths.get(fileName))
```

Call requires API level 26 (current min is 18): java.nio.file.Files#lines [more...](#) (Strg+F1)

```
return pupils
```

# EditPupilActivity

- Wieder mit `TextInputLayout`
  - Validierung
- FAB zur Speicherung
  - `DbTable`-Methoden
- Synchronisation über Datenbank

The screenshot shows two side-by-side mobile application screens. The left screen, titled 'Edit Pupil', is a form for editing a pupil's information. It has a blue header with the title. Below the header, there are four input fields: 'Lastname' with the value 'Hüttmaier', 'Firstname' with the value 'Franz', and 'City' with the value 'Vöcklamarkt'. Below these fields is a checkbox labeled 'Is male' which is checked. At the bottom right of the form is a pink circular floating action button (FAB) with a white document icon. The right screen, titled 'PupilManager', has a blue header with the title and a trash can icon and a dropdown arrow icon. Below the header is a list of pupils, each with a small blue icon of a person and their name: Böhlm Matthias, Eilmsteiner David, Garn Julian, Gubesch Christof, Harwöck Florian, Herzog Matthias, Hüttmaier Franz, and Kaindl Julian.

# Leeren der Datenbank

- Delete-Button leert DB
- Download leert DB zuerst und lädt dann pupils.csv neu

