



DB Battle: Realm VS Room

Tobias

Growth Session #22 - February 15 2019

Intro - General and Common ideas



ROOM



REALM

Intro - General and Common ideas



ROOM



REALM

Intro - General and Common ideas



ROOM

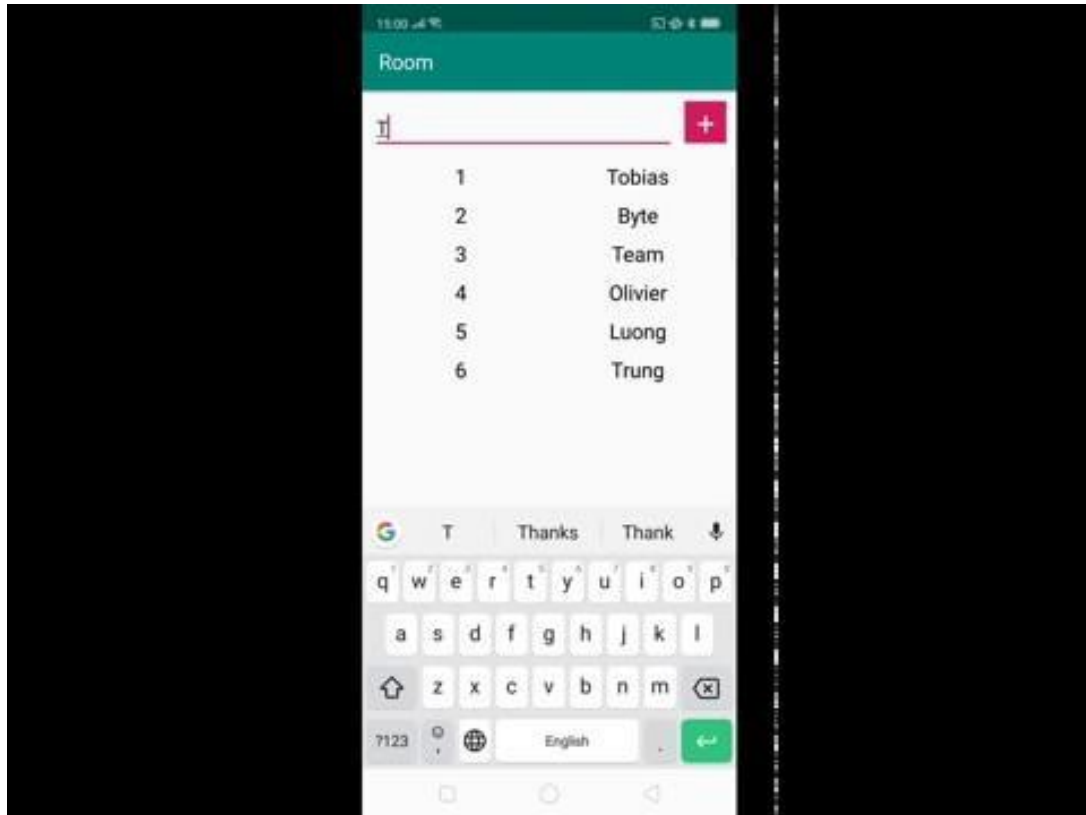
&&



REALM

-
- (Persistence) Libraries used for **data storage**, supports database-related stuff
 - **Read & write** data-objects, used in your app (Ex: Users)
 - Stored **locally** (= on the device) or **externally** (= on a server)
- **Local-storage** could be useful with caching OR lost internet-connection

Goals - Achievements



Intro - General Differences



ROOM



REALM



ROOM

VS



REALM

Intro - General Differences



ROOM

VS



REALM

Intro - General Differences



ROOM

VS



REALM



Intro - Own opinion



ROOM

VS



REALM



A bit more complicated to set up

- **Entity-class** (User with ID & Name)
→ Supports Auto-incrementation
- **Database-class** (Singleton = 1 instance)
- **Data Access Object-class** (Data-calls)

Very easy to set up

- **Entity-class** (User with ID & Name)
→ Manual Incrementation
- **Built-in :**

```
Realm.init(this)  
val realm = Realm.getDefaultInstance()
```

Intro - General Differences



ROOM

VS



REALM



- Mostly “**Self-defined**” methods

```
@Query("SELECT * FROM users")  
fun getAllUsers(): List<User>
```

- @Insert
- @Delete

- Mostly Built-in “**Chained**”-methods

```
realm.where(RealmUser::class.java).findAll()
```

- realm.insert(...)
- realm.delete(...)

CONCLUSION - PUBLIC OPINION



“Realm uses more RAM and increases the apk size & build time. So I prefer Room.”

- Alexey Zatsepin from Stack Overflow