UNIVERSITÀ
DEGLI STUDI
DE L'AQUILA

disim

Mobile Programming Laboratory

ANDROID
Storage II

Teachers

Ing. Tarquini Francesco, Ph.D

Ph.D in Computer Science Engineering

francesco.tarquini@univaq.it

Ing. D'Errico Leonardo

Ph.D Student in Computer Science Engineering

leonardo.derrico@graduate.univaq.it

Teaching Materials

Available on MOODLE platform

http://www.didattica.univaq.it

Google Drive Repository

https://drive.google.com/drive/folders/1ISqZfn0i9Ub3eWNXbvW00rd0hD9ya8OL?usp=sharing

Topics

- Room Database
  - Dependencies
  - Components
    - Entities
    - Dao
    - Database

Room Database

The Room persistence library provides an abstraction layer over SQLite to allow for more robust database access.

The library helps you create a cache of your app's data on a device that's running your app.

Google highly recommends using Room instead of SQLite.

It is presented on June 2017 during the Google I/O

Currently is available the version 1.1.1

The developer must include the dependency

```
implementation "android.arch.persistence.room:runtime:1.1.1"
```

Room Database - Dependencies

Android Studio projects are not configured to access the repository to default.

To add it to your project, open build.gradle file for your project and add the google() repository.

```
allprojects {
   repositories {
      google()
      jcenter()
   }
}
```

Now open the build.gradle file for your app or module and add

```
implementation "android.arch.persistence.room:runtime:1.1.1"
annotationProcessor "android.arch.persistence.room:compiler:1.1.1"
```

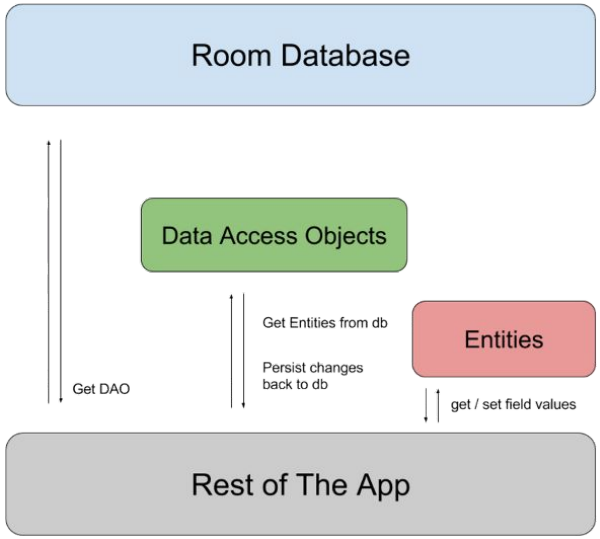
Room Database - Components

There are 3 major components in Room:

Database: contains the database holder and serves as the main access point for the underlying connection to your app's persisted, relational data.

Entity: represents a table within the database

DAO: contains the methods used for accessing the database



Room Database - Components

The Database components consist in a specific annotated class with **@Database** that should satisfy the following conditions:

Be an **abstract class** that extends **RoomDatabase**

Include the **list of entities** associated with the database within the annotation

Contain an **abstract method** that has 0 arguments and returns the class that is annotated with **@Dao**

At runtime, the developer can acquire an instance of Database by calling **Room.databaseBuilder()** or **Room.inMemoryDatabaseBuilder()**.

Room Database - Entities

The following code contains a sample database configuration.

```
@Entity(tableName=”cities”)
public class City {

  @PrimaryKey(autogenerate = true)
  private String id;

  @ColumnInfo(name = "city")
  private String city;

  @ColumnInfo(name = "region")
  private String region;

  // Get & Setter are required
}
```

Room Database - Dao

```
@Dao
public interface CityDao {

  @Query("SELECT * FROM cities")
  List<City> getAll();

  @Query("SELECT * FROM cities WHERE id IN (:cityIds)")
  List<City> loadAllByIds(int[] cityIds);

  @Query("SELECT * FROM cities WHERE city LIKE :city AND "
       + "region LIKE :region LIMIT 1")
  City findByName(String city, String region);

  @Insert(onConflict = OnConflictStrategy.REPLACE)
  void insertAll(City... cities);

  @Update
  void update(City city);

  @Delete
  void delete(City city);
}
```

Room Database - Database

```
@Database(entities = {City.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase {

   public abstract City cityDao();
}
```

After creating the files, the developer gets an instance of the created database using the following code

```
AppDatabase db = Room.databaseBuilder(getApplicationContext(), AppDatabase.class, "myDatabase").build();
```

Be Careful! The developer should follow the singleton pattern when instantiating an AppDatabase object.

Be Careful! The requests at the database must work in a separate thread from MainThread, but the developer can force it

```
AppDatabase db = Room.databaseBuilder(getApplicationContext(), AppDatabase.class, "myDatabase").allowMainThreadQueries().build();
```

Room Database - Database

All documentation is available on

https://developer.android.com/training/data-storage/room/