ЛЕКЦИЯ I

Android: Saving Data #1

Теодор Костадинов

Съдържание

Types of Storages

За записване на постоянни данни, в Андроид имаме различни варианти.

- App-specific Storage
  - App folder - всеки апп може да записва файлове във вътрешната директория на аппа. Тези файлове са лични и не могат да се достъпват от други аппове.
  - Shared Preferences - всеки апп може да създаде прости файлове от тип речник. Всяка данна се записва като ключ и стойност.
- Shared Storage - всеки апп може да създава файлове, които иска да сподели с други аппове.
- Databases - всеки апп може да създава бази данни. Те се съхраняват като файлове в личната папка на апп-а.

Storage Demo

Room Database

Room Database

```
dependencies {
  def room_version = "2.2.5"

  implementation "androidx.room:room-runtime:$room_version"
  annotationProcessor "androidx.room:room-compiler:$room_version"
```

Room Database

Библиотеката има три главни компонента:

- @Database — дефиниране на базата данни
- @Entity — дефиниране на ентити
- @Dao — репозитори

Setup

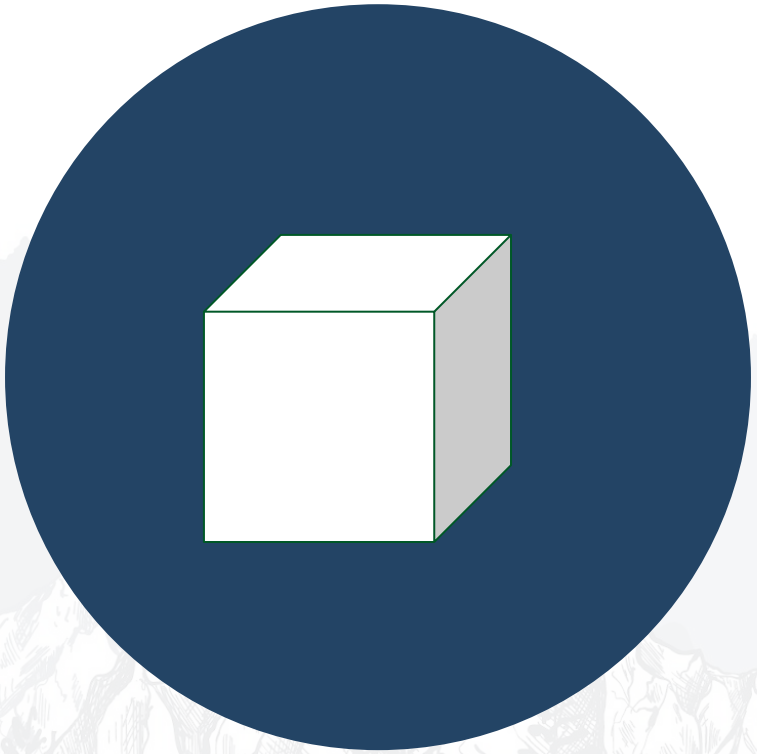
Setup

```
package com.my.app.data.local.database

@Database(entities = {}, version = 1)
public abstract class AppDatabase extends RoomDatabase {
    // TODO: add repositories
}
```

Setup

```
package com.vsc.myapplication.data.local.database;
public final class Database {
    private static AppDatabase instance;
    private Database() {}
    public static AppDatabase getInstance(Context context) {
        if (instance == null) {
            instance = Room.databaseBuilder(context, AppDatabase.class,
"database-name")
                    .fallbackToDestructiveMigration()
                    .build();
        }
        return instance;
    }
}
```



Entity

Entity

Описва какви данни ще записваме в базата данни в една таблица.
Три компонента:

- @Entity — задава класа като ентити
- @PrimaryKey — задава полето за индентификация
- @ColumnInfo

Entity

```
package com.my.app.data.local.database.users.models
@Entity(tableName = ”users”)
public class User {
    @PrimaryKey(autoGenerate = true)
    public int uid;
    @ColumnInfo(name = "first_name")
    public String firstName;
    @ColumnInfo(name = "last_name")
    public String lastName;

   // Getters and Setters
}
```

Entity

```
@Database(entities = {User.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase {
    // TODO: add repositories
}
```

Repository

Repository

Описва какви взаимодействия ще имаме с базата.

Четири компонента:

- @Dao задава класа като Data access object
- @Query(“”)
- @Insert
- @Delete

Repository

```
package com.my.app.data.local.database.users
@Dao
public interface UserDao {
    @Query("SELECT * FROM users")
    List<User> getAll();

    @Insert
    void insertAll(User... users);

    @Delete
    void delete(User user);
}
```

Repository

```
@Database(entities = {User.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase {
    public abstract UserDao userDao();
}
```

Service

Service

Room Database библиотеката ни ограничава да изпълняваме какъвто и да е код, който работи с базата данни, на UI нишката.
Обичайния подход е да си направим сървиз, който ще изпълнява този код на background нишка и ще връща резултатът чрез callback.

Service

```
private final UserDao userDao;
public UserService(Context context) {
    userDao = Database.getInstance(context).userDao();
}
public void getAllStudents(DataListener<List<Student>> dataListener) {
    new AsyncTask<Void, Void, List<Student>>() {
    @Override
    protected List<Student> doInBackground(Void... voids) {
        return userDao.getAll();
    }
    @Override
    protected void onPostExecute(List<Student> students) {
        dataListener.onData(students);
    }
    }.execute();
}
public interface DataListener<T> {
    void onData(T student);
}
```

ВЪПРОСИ?

HOMEWORK

Задача

Напишете програма, в която създавате списък с филми. Един филм трябва да има заглавие, година на излизане и оценка от 1 до 10.
Трябва да имате два екрана.
Един за създаване на нов филм. Другият за визуализиране на всички записани филми.

БЛАГОДАРЯ
ЗА ВНИМАНИЕТО!