Android LiveData

Anton Saatze

Motivation

Technical

Live Data

LiveData Docs LiveData in Cor

LiveData in Databinding

App Structu

General My App

Implementation

Summar

Android LiveData

Anton Saatze

University of applied Sciences Munich

22.may.2020

Table of Contents

- Motivation
- 2 Technical Background
- 3 LiveData

LiveData by the documentation LiveData in Code LiveData in Databinding

- 4 build LiveData into the Architecture General App Structure My small T-Shirt App LiveData in the T-Shirt App
- 5 Summary

LiveData in Code LiveData in Databinding

App Struct

My App

Implementation

Summary

Motivation

- core concept to present data
- clean way to hand data across the archictecture
- guarantee that latest data is presented in view
- first thing to learn, when joining a real android project team

Background

LiveData Docs LiveData in Code LiveData in Databinding

General My App

Implementation

Summar

Technical Background

- first thing to learn, when joining a real android project team
- 1/2 year internship in a android project
- learned android basics (Activty, Fragment, Lifecycle, Viewmodel, Views, LiveData, Databinding)
- my first task have been "redesign view xyz regarding to the designers wishes"

App Struct

My App

Implementatio

Summar

Technical Background

- learn how to design views with different layouts
- how to bind data to the view
- how adapter and lists work
- how to apply Data to a viewmodel
- how to publish Data by a viewmodel
- (Spoiler) its no magic, its LiveData

Motivation

Technical Backgroup

Live Data

LiveData In Code LiveData in

General

Implementation

impiementatio

LiveData is an observable data holder class. Unlike a regular observable, LiveData is lifecycle-aware, meaning it respects the lifecycle of other app components, such as activities, fragments, or services. This awareness ensures LiveData only updates app component observers that are in an active lifecycle state

developer.android.com (official documentaion of android)

Motivation

Technical Backgroup

Live Data

LiveData Docs
LiveData in Code
LiveData in

App Struc

My App Implementation

Implementatio

LiveData is an **observable data holder** class. Unlike a regular observable, LiveData is lifecycle-aware, meaning it respects the lifecycle of other app components, such as activities, fragments, or services. This awareness ensures LiveData only updates app component observers that are in an active lifecycle state

developer.android.com (official documentaion of android)

Motivation

Technical

Live Data

LiveData Docs

LiveData in Code LiveData in

LiveData in Databinding

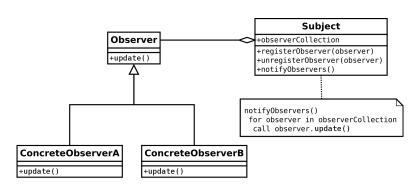
App Structu

General My App

Implementation

Summar

Observer Pattern



LiveData Docs

LiveData is an observable data holder class. Unlike a regular observable. LiveData is lifecycle-aware. meaning it respects the lifecycle of other app components, such as activities, fragments, or services. This awareness ensures LiveData only updates app component observers that are in an active lifecycle state

developer.android.com (official documentaion of android)

Summar

LifeCycle Awarenes

- Observer almost everytime need an LifeCycle
- only exception when using "observeForever" method
- LifeCycles are provided by the Activity and/or Fragments
- Observer only observe while the passed LifeCycle is (STARTED/RESUMED)
- Observer do not observe when the LifeCycle is PAUSED or in a dead state

Fazit: make sure, to have an lifecycle when you want to observe

Technical Background

LiveData Docs
LiveData in Code

LiveData in Code LiveData in Databinding

My App

Implementation

Create Mutablel iveData

```
val liveData = MutableLiveData<String>().apply{
value = "Hello World"
}
```

Modify MutableLiveData with operations like *postValue()* or *setValue()* (which only can be called from the main-thread)

Create LiveData

```
@Query("SELECT * FROM PERSON ORDER BY NAME")
LiveData<List<Person>> loadAllPersons();
```

LiveData in Code

My App

Create LiveData using MutableLiveData

```
private val privateLiveData =
MutableLiveData<String>().apply{
value = "Hello World"
}
val publicLiveData : LiveData<String> =
privateLiveData
fun changeString(string : String){
privateLiveData.postValue(string)
```

Live Data
LiveData Docs
LiveData in Code

LiveData in Code LiveData in Databinding

General My App

Implementatio

Summary

Code example

Observing LiveData with Observer object

Observing LiveData with Lambda

```
// not observed
val name = repository.publicLiveData
// make line above observing indirectly
val nameLetters = name.observe(
lifeCycleOwner,
Observer{it.toCharArray().asList()}
)
```

Motivation

Technical Background

Live Data
LiveData Docs
LiveData in Code

LiveData in Databinding

General My App

My App Implementation

Summary

name="model"

</data>

Observing Livedata + Databinding

```
Fragment
override fun onCreateView(
     inflater: LavoutInflater, container: ViewGroup?,
     savedInstanceState: Bundle?
 ): View? {
     val binding : FragmentDetailWithBindingBinding! = DataBindingUtil
          .inflate<FragmentDetailWithBindingBinding>(inflater,
             R.layout. fragment detail with binding,
              container,
              attachToParent: false)
     binding. model = viewModel
     binding. <u>lifecycleOwner</u> = this
     return binding. root
XML
<data>
   <variable</pre>
```

type="com.example.livedatapresentation.tshirt.screens.TShirtSelectorViewModel" />

LiveData in

Databinding

Observing Livedata + Databinding

XML < ImageView

```
android:id="@+id/t shirt image"
android:layout width="match parent"
android:layout height="wrap content"
android:scaleType="fitCenter"
android:src="@drawable/t shirt"
android:tint="@{model.tshirtColor}"
app:layout constraintLeft toLeftOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

Motivation

Technical

Backgrour

LiveData Docs LiveData in Code

LiveData in Databinding

Databinding

General My App

mpiementaek

Summary

Observing Livedata + Databinding

XML

```
<TextView
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_weight="0.5"
android:gravity="end"
android:text="@{model.selectedTShirt.size.toString()}"
android:textAlignment="textEnd"/>
```

</LinearLayout>

Motivation

Technical

Backgroun

Live Data LiveData Docs

LiveData in Code LiveData in Databinding

Ann Structu

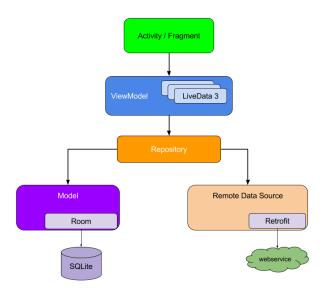
App Structi

General My App

Implementat

Summary

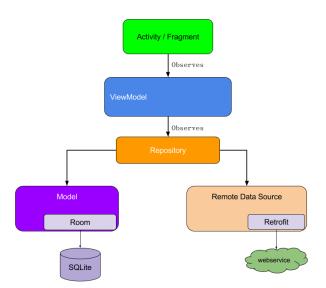
Overall Structure of Android Apps



LiveData in Code

General Mv App

Overall Structure of Android Apps



Motivation

Technical Background

Live Data
LiveData Docs
LiveData in Co

LiveData in Co LiveData in Databinding

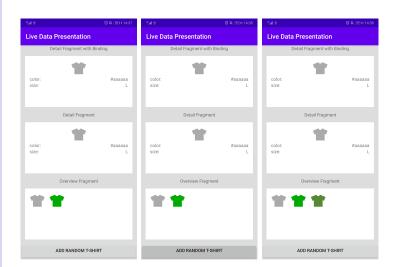
App Structure

My App

Implementation

Summary

T-Shirt App Fragments



Motivation

Technical Background

Live Data
LiveData Docs
LiveData in Coo

LiveData in Co LiveData in Databinding

App Structure

My App Implementation

....

T-Shirt App Fragments





Motivation

Technical

Backgroui

LiveData Docs LiveData in Code

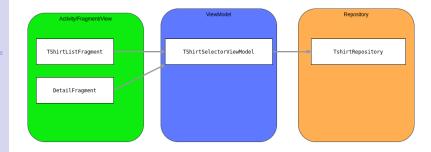
LiveData in Databinding

My App

mpiementatie

Summary

LiveData in the T-Shirt App



Motivation

Technical

Backgrou

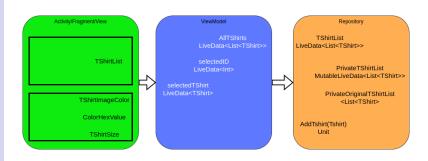
LiveData Docs
LiveData in Code
LiveData in

App Structur
General
My App

Implementation

Summary

LiveData in the T-Shirt App



Technical Background

Live Data
LiveData Docs
LiveData in Code
LiveData in
Databinding

General My App

Implementation

Summary

- check LifeCycle
 no observation when LifeCycle is Paused
- pass lifecycle to binding
- check if LiveData is observed not observed LiveData does not change
- check value is in proper Type in xml android:text = "@{viewmodel.number}" ≠ android:text = "@{viewmodel.number.toString()}"
- check Livedata is not Mutable acidently

Live Data
LiveData Docs
LiveData in Cod

App Structur

Му Арр

Implementation

Summary

more information

- udemy.com/course/android-jetpack/
- medium.com/androiddevelopers/viewmodels-and-livedatapatterns-antipatterns-21efaef74a54

developer.android.com:

- /topic/libraries/architecture/livedata
- /reference/androidx/lifecycle/Transformations
- /jetpack/docs/guide

Motivation

Technical Backgroun

Live Data

LiveData in Cod LiveData in Databinding

App Structure

My App Implementation

Summary

Thank you for your attention

Feel free to ask me anything about

- LiveData
- Android General
- work in the internship

Repo:

github.com/Tosaa/mobile-application-presentation