

Android

app

- manifests
- java
 - com.example.mvvmex1
 - adapter
 - dao
 - database
 - entity
 - repository
 - viewmodel
 - AddActivity
 - MainActivity
 - com.example.mvvmex1 (androidTest)
 - com.example.mvvmex1 (test)
- java (generated)
- res
 - drawable
 - layout
 - mipmap
 - values
 - xml
- res (generated)
- Gradle Scripts
 - build.gradle (Project: MVVM_Ex1)
 - build.gradle (Module: MVVM_Ex1.app)
 - gradle-wrapper.properties (Gradle Version: 4.10.1)
 - proguard-rules.pro (ProGuard Rules for Project)
 - gradle.properties (Project Properties)

User.kt x UserDatabase.kt x UserRepository.kt x UserViewModel.kt x UserAdapter.kt x AddActivity.kt x MainActivity.kt x

```
16 class MainActivity : AppCompatActivity() {
17
18     lateinit var userViewModel : UserViewModel
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         setContentView(R.layout.activity_main)
23
24         recyclerView : RecyclerView = findViewById(R.id.recyclerview)
25         recyclerView.layoutManager = LinearLayoutManager(context: this)
26         var userAdapter = UserAdapter()
27         recyclerView.adapter = userAdapter
28
29
30         val database = UserDatabase.getInstance(context: this)
31         val repository = UserRepository(database.userDao())
32         var userViewModelFactory = UserViewModelFactory(repository)
33
34         userViewModel = ViewModelProvider(owner: this, userViewModelFactory).get(UserViewModel::class.java)
35
36         userViewModel.users.observe(owner: this, Observer { users ->
37             Log.d(tag: " users count ", users.size.toString())
38             userAdapter.setDBUsers(users)
39         })
40     }
```

4 2

Android

app

manifests

java

com.example.mvvmex1

adapter

UserAdapter

dao

database

entity

repository

viewmodel

AddActivity

MainActivity

com.example.mvvmex1 (androidTest)

com.example.mvvmex1 (test)

java (generated)

res

drawable

layout

mipmap

values

xml

res (generated)

Gradle Scripts

build.gradle (Project: MVVM_Ex1)

build.gradle (Module: MVVM_Ex1.app)

gradle-wrapper.properties (Gradle Version)

User.kt

UserDatabase.kt

UserRepository.kt

UserViewModel.kt

UserAdapter.kt

UserDao.kt

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

```
override fun onCreateView(parent: ViewGroup, viewType: Int): UserViewHolder {  
    var view : View = LayoutInflater.from(parent.context)  
        .inflate(R.layout.user_item, parent, attachToRoot: false)  
    return UserViewHolder(view)  
}  
  
override fun onBindViewHolder(holder: UserViewHolder, position: Int) {  
    var user : User = users[position]  
    holder.title.text = user.userName  
    holder.pwd.text = user.pwd  
}  
  
override fun getItemCount(): Int {  
    return users.size  
}  
  
fun setDBUsers(users : List<User>){  
    Log.d(tag: "users count2", users.size.toString())  
    this.users = users  
    notifyDataSetChanged()  
}
```

Android

app

- manifests
- java
 - com.example.mvvmex1
 - adapter
 - UserAdapter
 - dao
 - database
 - entity
 - repository
 - viewmodel
 - AddActivity
 - MainActivity
 - com.example.mvvmex1 (androidTest)
 - com.example.mvvmex1 (test)
 - java (generated)
 - res
 - drawable
 - layout
 - mipmap
 - values
 - xml
 - res (generated)

Gradle Scripts

 - build.gradle (Project: MVVM_Ex1)
 - build.gradle (Module: MVVM_Ex1.app)

UserAdapter.kt

```
1 package com.example.mvvmex1.adapter
2
3 import android.util.Log
4 import android.view.LayoutInflater
5 import android.view.View
6 import android.view.ViewGroup
7 import android.widget.TextView
8 import androidx.cardview.widget.CardView
9 import androidx.recyclerview.widget.RecyclerView
10 import com.example.mvvmex1.R
11 import com.example.mvvmex1.entity.User
12
13 class UserAdapter: RecyclerView.Adapter<UserAdapter.UserViewHolder>() {
14
15     var users : List<User> = ArrayList()
16
17     class UserViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){
18
19         val title : TextView = itemView.findViewById(R.id.username)
20         var pwd : TextView = itemView.findViewById(R.id.pwd)
21         var cardView : CardView = itemView.findViewById(R.id.cardview)
22
23     }
24 }
```

Android

app

- manifests
- java
 - com.example.mvvmex1
 - adapter
 - dao
 - database
 - entity
 - repository
 - viewmodel
 - UserViewModel.kt
 - com.example.mvvmex1 (androidTest)
 - com.example.mvvmex1 (test)
- java (generated)
- res
 - drawable
 - layout
 - mipmap
 - values
 - xml
- res (generated)

Gradle Scripts

- build.gradle (Project: MVVM_Ex1)
- build.gradle (Module: MVVM_Ex1.app)
- gradle-wrapper.properties (Gradle Version)
- proguard-rules.pro (ProGuard Rules for MVVM_

User.kt x UserDatabase.kt x UserRepository.kt x UserViewModel.kt x UserDao.kt x

```
1 package com.example.mvvmex1.viewmodel
2
3 import android.util.Log
4 import androidx.lifecycle.LiveData
5 import androidx.lifecycle.ViewModel
6 import androidx.lifecycle.ViewModelProvider
7 import androidx.lifecycle.asLiveData
8 import com.example.dbexample1.repository.UserRepository
9 import com.example.mvvmex1.entity.User
10
11 class UserViewModel(private val userRepository : UserRepository) : ViewModel() {
12
13     var users : LiveData<List<User>> = userRepository.getUsers().asLiveData()
14 }
15
16 class UserViewModelFactory(private val userRepository : UserRepository) : ViewModelProvider.Factory{
17     override fun <T : ViewModel> create(modelClass: Class<T>): T {
18         if(modelClass.isAssignableFrom(UserViewModel::class.java)){
19             Log.d( tag: " users count ", msg: "UserViewModelFactory: ")
20             return UserViewModel(userRepository) as T
21         }else{
22             throw java.lang.IllegalArgumentException("Un-known view model")
23         }
24     }
25 }
```


Android

app

manifests

java

com.example.mvvmex1

adapter

dao

database

entity

repository

UserRepository

viewmodel

AddActivity

MainActivity

com.example.mvvmex1 (androidTest)

com.example.mvvmex1 (test)

java (generated)

res

drawable

layout

mipmap

values

xml

res (generated)

Gradle Scripts

build.gradle (Project: MVVM Ex1)

User.kt

UserDatabase.kt

UserRepository.kt

UserDao.kt

```
1 package com.example.dbexample1.repository
2
3
4 import android.util.Log
5 import com.example.dbexample1.dao.UserDao
6 import com.example.mvvmex1.entity.User
7 import kotlinx.coroutines.flow.Flow
8
9 class UserRepository(private val userDao : UserDao) {
10
11     fun getUsers() : Flow<List<User>> {
12         Log.d( tag: " users count ", msg: "getUsers repo : ")
13         return userDao.getAllUsers()
14     }
15
16     //var users : Flow<List<User>> = userDao.getAllUsers()
17
18     fun insertUser(user : User) {
19         userDao.insert(user)
20     }
21 }
```

app > src > main > java > com > example > mvvmex1 > database > FirebaseDatabase > companion object

Android | Nexus 5X API 30

User.kt | FirebaseDatabase.kt | UserDao.kt

Project

- app
 - manifests
 - java
 - com.example.mvvmex1
 - adapter
 - dao
 - database
 - entity
 - User
 - repository
 - viewmodel
 - com.example.mvvmex1 (androidTest)
 - com.example.mvvmex1 (test)
 - java (generated)
 - res
 - drawable
 - layout
 - mipmap
 - values
 - xml
 - res (generated)
 - Gradle Scripts
 - build.gradle (Project: MVVM_Ex1)
 - build.gradle (Module: MVVM_Ex1.app)
 - gradle-wrapper.properties (Gradle Version)
 - proguard-rules.pro (ProGuard Rules for MVVM_E

Resource Manager

Structure

Bookmarks

Build Variants

```
3 import android.content.Context
4 import androidx.room.Database
5 import androidx.room.Room
6 import androidx.room.RoomDatabase
7 import com.example.dbexample1.dao.UserDao
8 import com.example.mvvmex1.entity.User
9
10 @Database(entities = [User::class], version = 1)
11 abstract class FirebaseDatabase : RoomDatabase() {
12
13     abstract fun userDao() : UserDao
14
15     companion object {
16         private var instance : FirebaseDatabase? = null
17         @Synchronized
18         fun getInstance(context : Context) : FirebaseDatabase {
19             if(instance == null){
20                 instance = Room.databaseBuilder(context.applicationContext,
21                     FirebaseDatabase::class.java, name: "user_tb")
22                     .allowMainThreadQueries()
23                     .build()
24             }
25             return instance!!
26         }
27     }
28 }
```

Android

app

- manifests
- java
 - com.example.mvvmex1
 - adapter
 - dao
 - UserDao
 - database
 - entity
 - repository
 - viewmodel
 - AddActivity
 - MainActivity
 - com.example.mvvmex1 (androidTest)
 - com.example.mvvmex1 (test)
 - java (generated)
 - res
 - drawable
 - layout
 - mipmap
 - values
 - xml
 - res (generated)
- Gradle Scripts
 - build.gradle (Project: MVVM/Ex1)

User.kt x FirebaseDatabase.kt x UserDao.kt x

```
1 package com.example.dbexample1.dao
2
3 import androidx.room.Dao
4 import androidx.room.Insert
5 import androidx.room.Query
6 import com.example.mvvmex1.entity.User
7 import kotlinx.coroutines.flow.Flow
8
9 @Dao
10 interface UserDao {
11
12     @Insert
13     fun insert(user : User)
14
15     @Query("Select * from user_tb where userName = :userName and pwd = :pwd ")
16     fun getUser(userName : String, pwd: String) : User
17
18     @Query("select * from user_tb")
19     fun getAllUsers() : Flow<List<User>>
20
21 }
```

Android

app

manifests

java

com.example.mvvmex1

adapter

dao

database

entity

User

repository

viewmodel

AddActivity

MainActivity

com.example.mvvmex1 (androidTest)

com.example.mvvmex1 (test)

java (generated)

res

drawable

User.kt

UserDatabase.kt

UserDao.kt

```
1 package com.example.mvvmex1.entity
2
3 import androidx.room.Entity
4 import androidx.room.PrimaryKey
5
6 @Entity(tableName = "user_tb")
7 class User (var userName : String,
8             var pwd : String,) {
9     @PrimaryKey(autoGenerate = true)
10     var id = 0
11 }
```


dActivity.kt × User.kt × AndroidManifest.xml × UserViewModel.kt × MainActivity.kt × activity_main.xml × user_item.xml ×

Code Split Design

Palette

Common

Ab TextView

Button

ImageView

RecyclerView

FragmentContainerView

ScrollView

Switch

Component Tree

cardview

LinearLayout (vertical)

Ab username "TextView"

Ab pwd "TextView"

user_item.xml

Pixel

12:00

MVVM Ex1

TextView

TextView

View

Attributes

LinearLayout

id

Declared Attributes

layout_width

match_parent

layout_height

wrap_content

layout_margin

10dp

orientation

vertical

Layout

layout_width

match_parent

layout_height

wrap_content

visibility

visibility

Transforms

dActivity.kt × User.kt × AndroidManifest.xml × UserViewModel.kt × MainActivity.kt × activity_main.xml × user_item.xml ×

Code Split Design

Palette

Common

Ab TextView

Button

ImageView

RecyclerView

FragmentContainerView

ScrollView

Switch

Component Tree

ConstraintLayout

recyclerview

activity_main.xml

Pixel

0dp

12:00

MVVM Ex1

TextView

TextView

TextView

TextView

TextView

TextView

TextView

TextView

Attributes

RecyclerView

recyclerview

id

recyclerview

Declared Attributes

layout_width

match_parent

layout_height

wrap_content

id

recyclerview

listitem

@layout/user_item

Layout

Constraint Widget

+

+

+

+

+

+

+

+

Constraints (0)

layout_width

match_parent