

`androidx.navigation.fragment.ktx`: Библиотека для работы с навигацией между фрагментами в приложении. Упрощает передачу данных и управление фрагментами.

`androidx.lifecycle.viewmodel.ktx`: предоставляет компоненты для управления жизненным циклом приложений с использованием `ViewModel`. Упрощает работу с данными, которые должны пережить уничтожение активности или фрагмента.

`androidx.activity.ktx`: Библиотека, упрощающая работу с компонентами активности. Включает расширения и утилиты.

`androidx.lifecycle.lifecycle.viewmodel.ktx`: Библиотека с расширениями для `ViewModel`, которая помогает управлять состоянием UI в зависимости от жизненного цикла.

`androidx.core.ktx`: Основные утилиты, которые делают работу с Android API более удобной благодаря расширениям.

`androidx.appcompat`: Библиотека, обеспечивающая совместимость старых версий Android, а также предоставляющая доступ к компонентам Material Design.

`com.google.android.material`: Библиотека компонентов Material Design для создания современных пользовательских интерфейсов.

`androidx.activity`: Библиотека, содержащая современные классы для работы с активностями, предоставляющая дополнительные функции и улучшения.

`androidx.constraintlayout`: Библиотека для управления макетом с использованием `ConstraintLayout`, позволяющая создать более сложные интерфейсы без глубокой вложенности.

`junit`: Библиотека для написания модульных тестов на Java. Широко используется в Android-разработке.

`androidx.espresso.core`: Библиотека для UI-тестирования, позволяющая автоматизировать взаимодействие с пользовательским интерфейсом.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/main"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".MainActivity">
10
11     <TextView
12         android:id="@+id/textView"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:text="Hello World!"
16         app:layout_constraintBottom_toBottomOf="parent"
17         app:layout_constraintEnd_toEndOf="parent"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent" />
20
21     <FrameLayout
22         android:id="@+id/frame1"
23         android:layout_width="409dp"
24         android:layout_height="354dp"
25         android:layout_marginStart="1dp"
26         android:layout_marginTop="1dp"
27         android:layout_marginBottom="1dp"
28         app:layout_constraintBottom_toTopOf="@+id/textView"
29         app:layout_constraintStart_toStartOf="parent"
30         app:layout_constraintTop_toTopOf="parent">
31     </FrameLayout>
32
33     <FrameLayout
34         android:id="@+id/frame2"
35         android:layout_width="409dp"
36         android:layout_height="354dp"
37         android:layout_marginTop="1dp"
```

```
16         app:layout_constraintBottom_toBottomOf="parent"
17         app:layout_constraintEnd_toEndOf="parent"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent" />
20
21     <FrameLayout
22         android:id="@+id/frame1"
23         android:layout_width="409dp"
24         android:layout_height="354dp"
25         android:layout_marginStart="1dp"
26         android:layout_marginTop="1dp"
27         android:layout_marginBottom="1dp"
28         app:layout_constraintBottom_toTopOf="@+id/textView"
29         app:layout_constraintStart_toStartOf="parent"
30         app:layout_constraintTop_toTopOf="parent">
31     </FrameLayout>
32
33     <FrameLayout
34         android:id="@+id/frame2"
35         android:layout_width="409dp"
36         android:layout_height="354dp"
37         android:layout_marginTop="1dp"
38         android:layout_marginEnd="1dp"
39         app:layout_constraintBottom_toBottomOf="parent"
40         app:layout_constraintEnd_toEndOf="parent"
41         app:layout_constraintTop_toBottomOf="@+id/textView">
42     </FrameLayout>
43 </androidx.constraintlayout.widget.ConstraintLayout>
```

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" ⚠ 1 4
2   ⚡ android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:orientation="vertical">
5
6     <Button
7         android:id="@+id/button_send"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Send Data" />
11 </LinearLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" ⚠ 1 ⚠ 1 ^
   ⚡ android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:orientation="vertical">

     <TextView
         android:id="@+id/textView_received"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Received Data" />
</LinearLayout>
```

blankfragment


```
1 package com.example.a10_android
2
3 > import ...
10
11 private const val ARG_PARAM1 = "param1"
12 private const val ARG_PARAM2 = "param2"
13
14 </> class BlankFragment : Fragment() {
15     private lateinit var sharedViewModel: SharedViewModel
16
17     override fun onCreateView(
18         inflater: LayoutInflater, container: ViewGroup?,
19         savedInstanceState: Bundle?
20     ): View? {
21         sharedViewModel = ViewModelProvider(requireActivity()).get(SharedViewModel::class.java)
22         val view = inflater.inflate(R.layout.fragment_blank, container, attachToRoot: false)
23
24         val button = view.findViewById<Button>(R.id.button_send)
25         button.setOnClickListener {
26             sharedViewModel.selectedData.value = "Hello from BlankFragment"
27         }
28
29         return view
30     }
31 }
```

Blankfragment2

```
1 package com.example.a10_android
2
3 > import ...
11
12 private const val ARG_PARAM1 = "param1"
13 private const val ARG_PARAM2 = "param2"
14
15 </> class BlankFragment2 : Fragment() {
16     private lateinit var sharedViewModel: SharedViewModel
17
18     override fun onCreateView(
19         inflater: LayoutInflater, container: ViewGroup?,
20         savedInstanceState: Bundle?
21     ): View? {
22         sharedViewModel = ViewModelProvider(requireActivity()).get(SharedViewModel::class.java)
23         val view = inflater.inflate(R.layout.fragment_blank2, container, attachToRoot: false)
24
25         sharedViewModel.selectedData.observe(viewLifecycleOwner, Observer { data ->
26             val textView = view.findViewById<TextView>(R.id.textView_received)
27             textView.text = data
28         })
29
30         return view
31     }
32 }
```

Mainactivity

```
1 package com.example.a10_android
2
3 > import ...
11
12 class MainActivity : AppCompatActivity() {
13     lateinit var binding: ActivityMainBinding
14     private lateinit var sharedViewModel: SharedViewModel
15
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)
18         binding = ActivityMainBinding.inflate(layoutInflater)
19         setContentView(binding.root)
20
21         sharedViewModel = ViewModelProvider( owner: this).get(SharedViewModel::class.java)
22
23         openFragment(BlankFragment(), R.id.frame1)
24         openFragment(BlankFragment2(), R.id.frame2)
25     }
26
27     private fun openFragment(f: Fragment, idHolder: Int) {
28         supportFragmentManager
29             .beginTransaction()
30             .replace(idHolder, f)
31             .commit()
32     }
33 }
```

 Edit translations for all locales in the translations editor.

```
1 <resources>
2     <string name="app_name">a10_android</string>
3     <!-- TODO: Remove or change this placeholder text -->
4     <string name="hello_blank_fragment">Hello blank fragment</string>
5 </resources>
```

Send Data

Hello World!

Hello from BlankFragment