

SAMSUNG



Kotlin

Базовый Курс

3.8. Навигация в Android приложениях



Виды навигации

SAMSUNG



- Иерархическая навигация
- Боковая навигация
- Навигация по стеку возврата
- Глубокие ссылки





Унификация навигации, примеры

SAMSUNG



- кнопка "домой" - должна всегда приводить на один и тот же начальный экран,
- кнопка "назад" - делает строго обратный по хронологии (back stack) переход по экранам, вне зависимости от их иерархии и при достижении начала цепочки может позволять выход из приложения,
- кнопка "вверх", должна поднимать пользователя по иерархии, но не должна делать выход из приложения, когда достигнут верхний уровень,
- свайп используется либо для перелистывания вкладок либо для перелистывания master-detail окон,
- использовать нестандартные управляющие элементы нежелательно, как например двойной или тройной клик.





С точки зрения разработчика



- По цели навигации.
 - переход на новый фрагмент,
 - переход на новую активность,
 - переход на другое приложение.
- По элементам, обеспечивающих навигацию.
 - кликабельные UI элементы (Button, ImageButton, пункты списков и т.д.),
 - навигационные UI элементы (меню активности, боковая панель Drawer, TopAppBar, ToolBar, BottomNavigation),
 - свайпы,
 - управление стеком возврата.





Navigation API, преимущества



- создание единого графического представление всего графа переходов,
- контроль переходов в приложении на соответствии построенной схеме,
- унификация и упрощения перехода к новой активности или фрагменту, а также передаче им данных,
- модифицированные навигационные элементы UI теперь интегрированы с новым механизмом навигации, вследствие чего нужно писать значительно меньше кода при их реализации,
- упрощение описания deep link,
- засчет упрощения программирования переходов разрабатывать сложное приложение станет легче и быстрее, особенно это заметно в случае если нужно быстро переделать схему переходов.





▶ Пример, build.gradle(app)

SAMSUNG



```
def nav_version = "2.3.0-alpha05"
```



```
implementation "androidx.navigation:navigation-fragment-ktx:$nav_version"
```

```
implementation "androidx.navigation:navigation-ui-ktx:$nav_version"
```

```
// Dynamic Feature Module Support
```

```
implementation "androidx.navigation:navigation-dynamic-features-fragment:$nav_version"
```





Пример, activity_main.xml

SAMSUNG



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_graph" />
</androidx.constraintlayout.widget.ConstraintLayout>
```





▶ Пример, MainActivity.kt

SAMSUNG



```
class MainActivity : AppCompatActivity() {  
    ▶ override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```





▶ Пример, BlankFragment.kt

```
class BlankFragment : Fragment() {
```



```
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {
```

```
        var view:View=inflater.inflate(R.layout.fragment1, container, false)
```

```
        view.findViewById<Button>(R.id.button5).setOnClickListener(Navigation.createNavigateOnClickListener(R.id.action_blankFragment_to_blankFragment4))
```

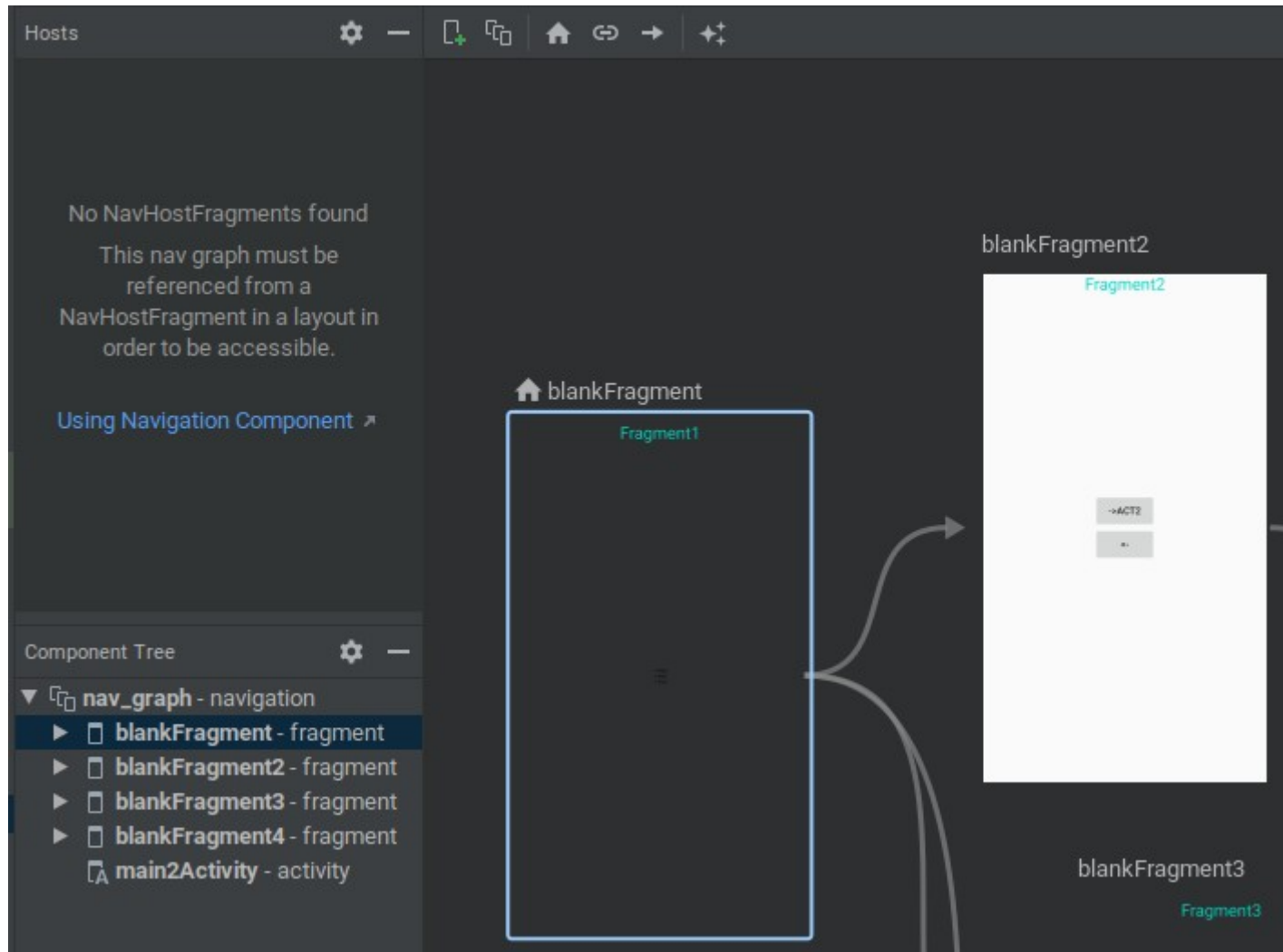
```
        return view
```

```
    }
```





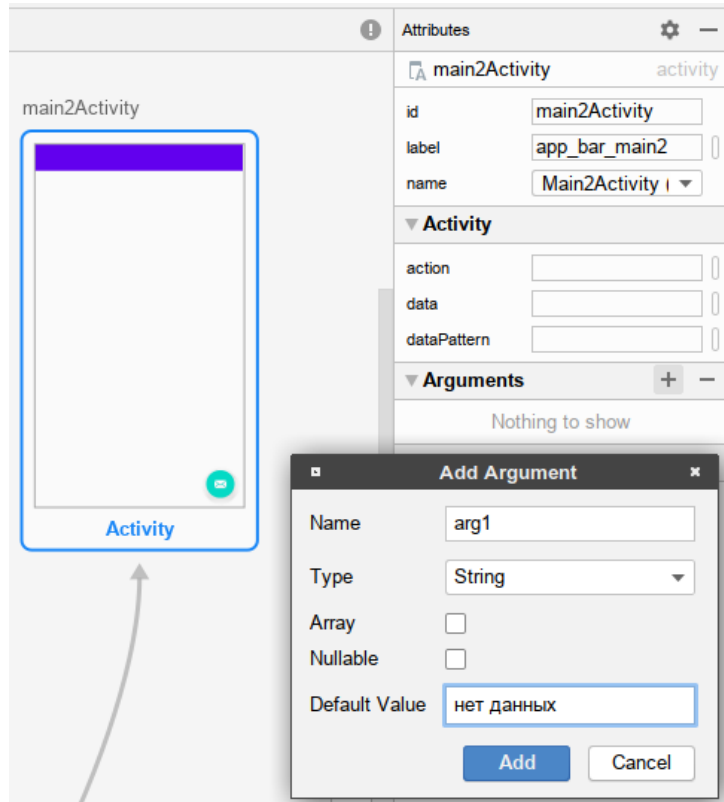
Пример, nav_graph.xml





Передача параметров

SAMSUNG



- передача

```
btn6.setOnClickListener(View.OnClickListener { view ->  
    // создаем бандл  
    val bundle:Bundle=Bundle()  
    // вносим в него значения  
    bundle.putString("arg1",et.text.toString())  
    // выполняем навигацию с передачей данных  
    view.findNavController().navigate(  
        R.id.action_blankFragment4_to_main2Activity,bundle)  
    })
```

- прием

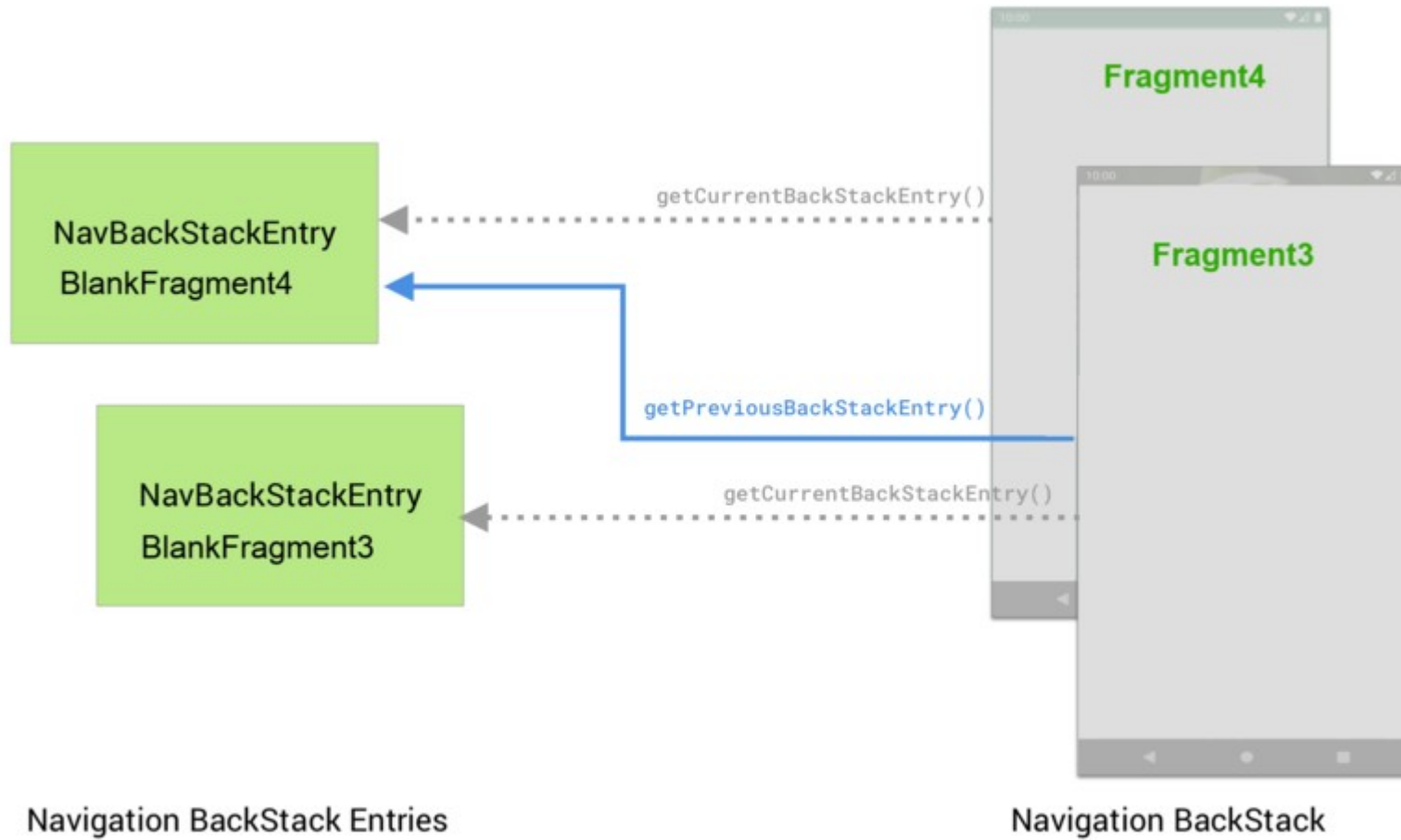
```
val str=intent.getStringExtra("arg1")  
val str=arguments.getString("arg1")
```





Возврат значений

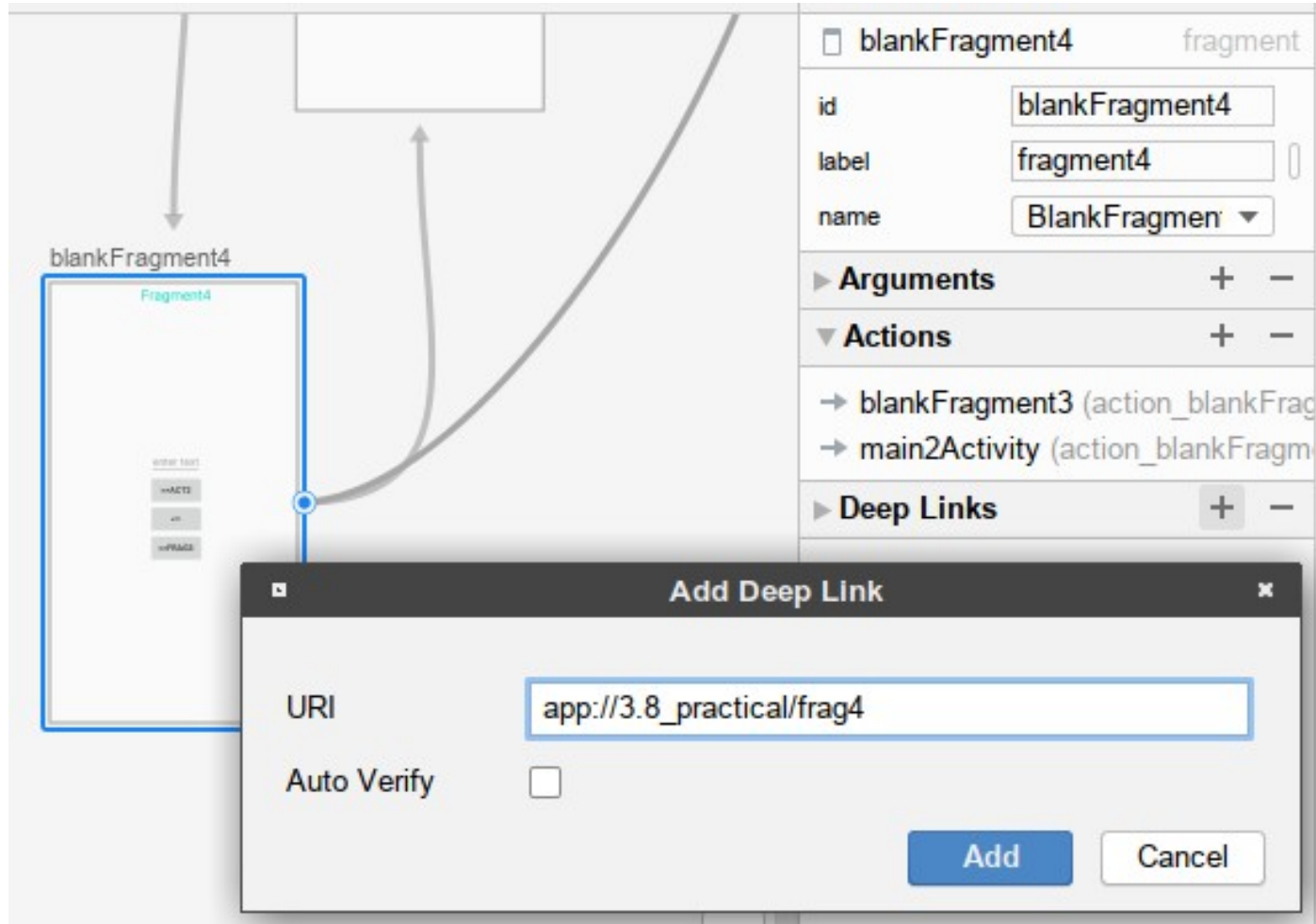
SAMSUNG





Deep Link

SAMSUNG





Global Action

SAMSUNG



Attributes

blankFragment4

Fragment4

enter text

enter

exit

pop

action_global_blankFragment4

id: al_blankFragment4

destination: blankFragment

Animations

enterAnim

exitAnim

popEnterAnim

popExitAnim

Argument Default Values

Pop Behavior

popUpTo: none

popUpToInclu...

Launch Options

launchSingleTop





Back vs Up

SAMSUNG





SAMSUNG



Спасибо!

