

برنامه نویسی دستگاه های سیار (CE364)

جلسه دهم: معرفی قطعه

سجاد شیرعلی شمرضا

پاییز 1401

شنبه، 19 آذر 1401

● بخشهای مرتبط با این جلسه:

- Unit 3: Navigation:
 - Pathway 2: Introduction to the Navigation component



سوال؟

قطعه



No one is on your speed dial yet

[ADD A FAVORITE](#)



Favorites



Recents



Contacts

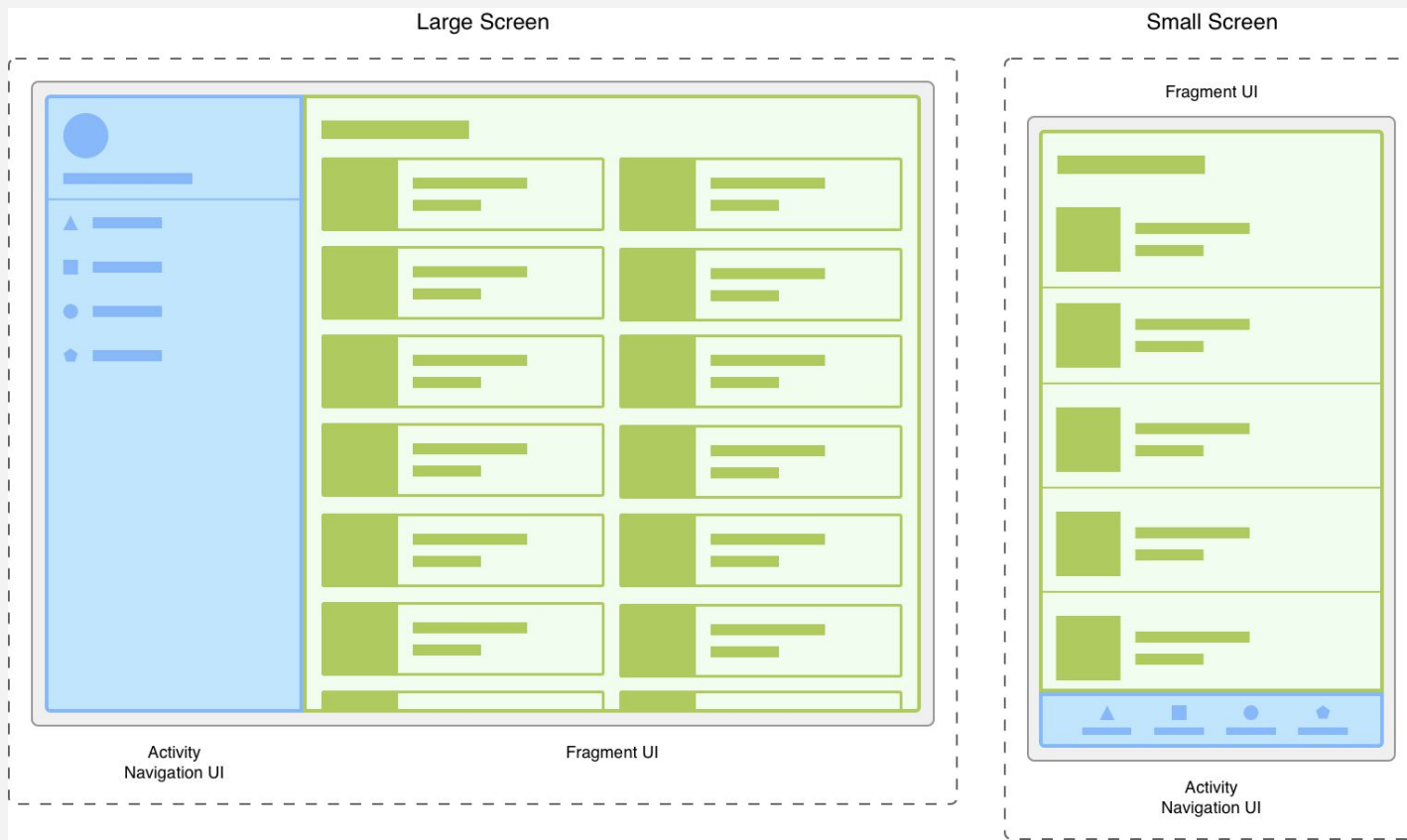


Voicemail

قطعه

- قصد: یک روش برای رفتن از یک فعالیت به فعالیت دیگر
- لزوما نیازی به نمایش فعالیت جدید نیست
- صرفا نیاز به تغییر بخشی از محتوای صفحه
- قطعه (Fragment): بخش قابل استفاده مجدد از رابط کاربری
- تعویض یک قطعه با قطعه دیگر
- استفاده از یک قطعه در فعالیت های مختلف

نمایش چند قطعه با هم



چرخه زندگی یک قطعه

- 5 حالت مختلف (Lifecycle.State):

- مقداردهی اولیه شده (INITIALIZED): یک نمونه از آن مقداردهی اولیه شده است
- ساخته شده (CREATED): اولین تابع چرخه زندگی اجرا شده و نمای متناظر ساخته شده است
- آغاز شده (STARTED): قطعه قابل مشاهده است، اما تمرکز بر روی آن نیست
- از سرگرفته شده (RESUMED): قطعه قابل مشاهده است و تمرکز بر روی آن است
- از بین رفته (DESTROYED): نمونه قطعه از بین برده شده است

توابع مربوط به تغییر حالت

- پس از ایجاد شی و ورود به حالت CREATED: تابع onCreate
- پس از ایجاد شی و برای ساخت نمای متناظر (حالت CREATED): تابع onCreateView
- پس از ایجاد نما برای شی: onViewCreated
- هنگامی که شی به حالت STARTED وارد می شود: onStart
- هنگامی که شی تمرکز پیدا می کند و وارد حالت RESUMED می شود: onResume
- بازگشت به حالت STARTED در حالیکه هنوز قابل مشاهده است: onPause
- بازگشت مجدد به حالت CREATED در حالیکه دیگر شی دیگر قابل مشاهده نیست: onStop
- قبل از ورود به حالت DESTROYED و موقعی که نما آزاد شده است: onDestroyView
- موقعی که شی وارد حالت DESTROYED می شود: onDestroy

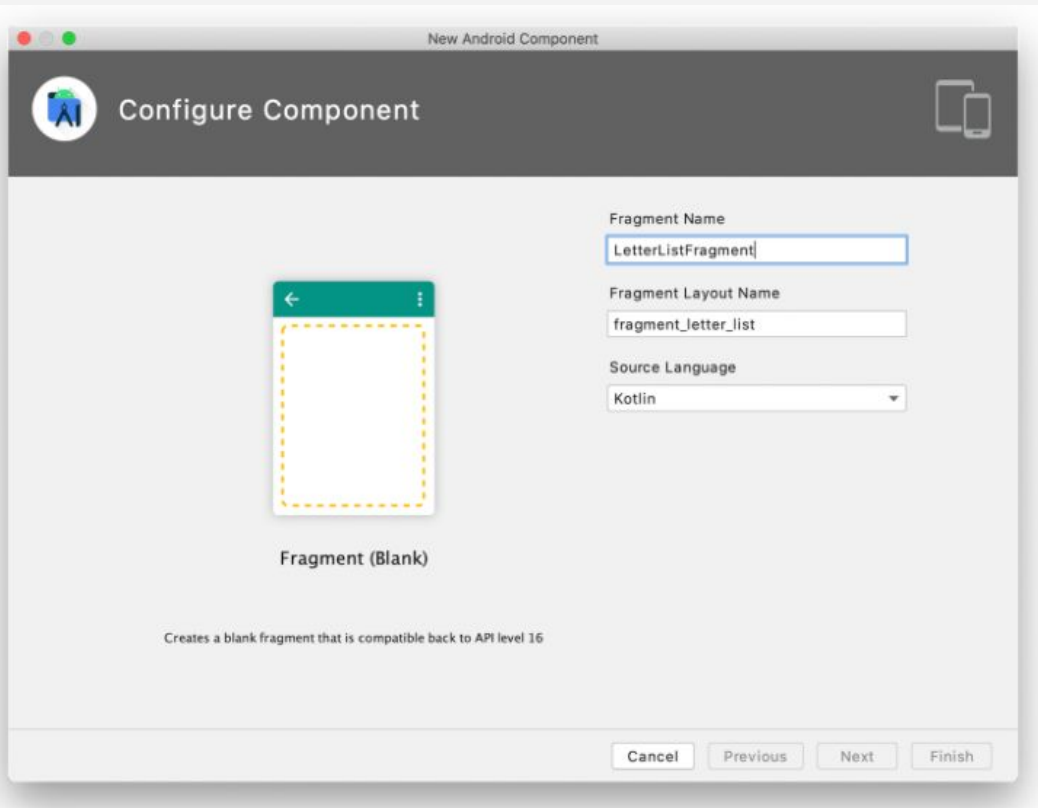
چرخه زندگی یک قطعه

Lifecycle State	Callback
CREATED	onCreate()
	onCreateView()
	onViewCreated()
STARTED	onStart()
RESUMED	onResume()
STARTED	onPause()
CREATED	onStop()
	onDestroyView()
DESTROYED	onDestroy()

تفاوت onCreate در قطعه و فعالیت

- در فعالیت: محل ایجاد و پر کردن (inflate) نما
- در قطعه: هنوز نما ایجاد نشده است
 - نما در onCreateView ایجاد و پر می شود
 - پس از ایجاد نما، onViewCreated اجرا می شود

اضافه کردن یک قطعه



- هر قطعه دو فایل متناظر دارد:
 - تعریف چیدمان در یک XML
 - کلاس متناظر برای کنترل
- اضافه از طریق
 - File > New > Fragment
 - > Fragment-Blank

تعریف قطعه برای برنامه لغتنامه

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LetterListFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:clipToPadding="false"
        android:padding="16dp" />

</FrameLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".WordListFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:clipToPadding="false"
        android:padding="16dp"
        tools:listitem="@layout/item_view" />

</FrameLayout>
```

کلاس اتصال (Binding Class)

- تولید خودکار توسط اندروید استودیو
 - با فعال کردن ویژگی viewBinding در buildFeatures فایل build.gradle

```
private var _binding: FragmentLetterListBinding? = null
```

- میتواند null باشد (در بازه ساخت قطعه و موقع ایجاد چیدمان)
- نمای یک قطعه ممکن است چندین بار ساخته و از بین برده شود

```
private val binding get() = _binding!!
```

توابع ایجاد قطعه

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setHasOptionsMenu(true)  
}  
  
override fun onCreateView(  
    inflater: LayoutInflater, container: ViewGroup?,  
    savedInstanceState: Bundle?  
): View? {  
    _binding = FragmentLetterListBinding.inflate(inflater, container, false)  
    val view = binding.root  
    return view  
}  
  
private lateinit var recyclerView: RecyclerView  
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
    recyclerView = binding.recyclerView  
    chooseLayout()  
}  
override fun onDestroyView() {  
    super.onDestroyView()  
    _binding = null  
}
```

ایجاد منو

```
override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {  
    inflater.inflate(R.menu.layout_menu, menu)  
  
    val layoutButton = menu.findItem(R.id.action_switch_layout)  
    setIcon(layoutButton)  
}
```

بقیه توابع

```
private fun chooseLayout() {
    when (isLinearLayoutManager) {
        true -> {
            recyclerView.layoutManager = LinearLayoutManager(context)
            recyclerView.adapter = LetterAdapter()
        }
        false -> {
            recyclerView.layoutManager = GridLayoutManager(context, 4)
            recyclerView.adapter = LetterAdapter()
        }
    }
}

private fun setIcon(menuItem: MenuItem?) {
    if (menuItem == null)
        return

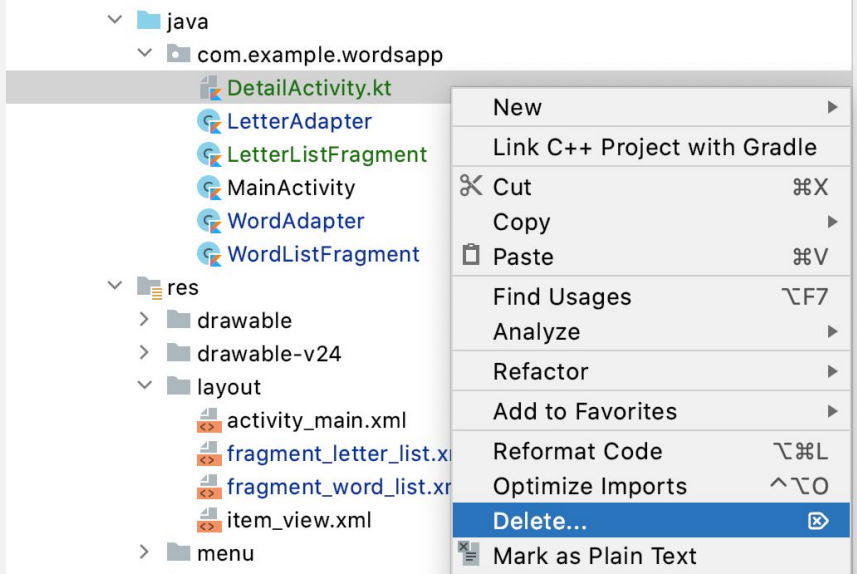
    menuItem.icon =
        if (isLinearLayoutManager)
            ContextCompat.getDrawable(this.requireContext(), R.drawable.ic_grid_layout)
        else ContextCompat.getDrawable(this.requireContext(), R.drawable.ic_linear_layout)
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_switch_layout -> {
            isLinearLayoutManager = !isLinearLayoutManager
            chooseLayout()
            setIcon(item)

            return true
        }
        else -> super.onOptionsItemSelected(item)
    }
}
```

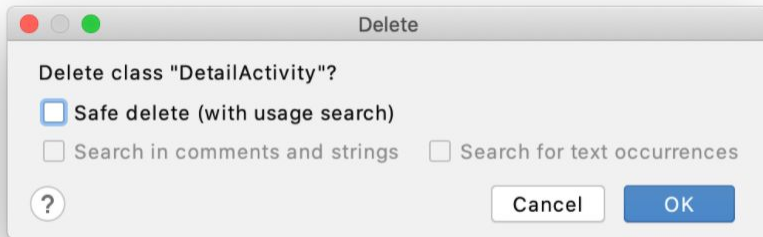

نمایش قطعه در فعالیت اصلی

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    val binding = ActivityMainBinding.inflate(layoutInflater)  
    setContentView(binding.root)  
}
```



حذف DetailActivity

- حذف فایل XML و کلاس کاتلین
- به روز رسانی AndroidManifest.xml



```
<activity
    android:name=".DetailActivity"
    android:parentActivityName=".MainActivity" />
```



سوال؟

تعويض قطعه

مدیریت قطعات

- استفاده از بخش ناوبری (Navigation) مجموعه جت پک (Android Jetpack)
- داری سه بخش اصلی:
 - گراف ناوبری: یک فایل XML برای تعریف مقاصد (Destination) و چگونگی حرکت بین آنها (که با نام عمل یا action شناخته می شوند)
 - میزبان (NavHost): یک مقصد را نمایش می دهد
 - کنترلر (NavController): کنترل چگونگی انتقال از یک مقصد به مقصد دیگر

پیش نیاز

- تعیین حداقل نسخه ناوبری در build.gradle، در بخش buildscript > ext
- اضافه کردن نیازمندی ها

```
buildscript {  
    ext {  
        appcompat_version = "1.2.0"  
        constraintlayout_version = "2.0.2"  
        core_ktx_version = "1.3.2"  
        kotlin_version = "1.3.72"  
        material_version = "1.2.1"  
        nav_version = "2.3.1"  
    }  
  
    ...  
}
```

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

انتقال اطلاعات بین قطعات

- پلاگین Safe Args برای انتقال اطلاعات بین قطعات
- اضافه کردن پیش نیاز در بخش dependencies buildscript > build.gradle فایل

```
classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"
```

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
    id 'kotlin-kapt'  
    id 'androidx.navigation.safeargs.kotlin'  
}
```

- اضافه کردن پلاگین در بخش

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

[Sync Now](#)

[Ignore these changes](#)

ساختار گراف ناوبری

- تعریف مقاصد مختلف در یک گراف ناوبری (Navigation Graph یا NavGraph)
- تعریف به صورت یک فایل XML
- یک نمونه از کلاس NavGraph ایجاد خواهد شد
- قطعات در یک FragmentContainerView نمایش داده می شوند

به روزرسانی فعالیت اصلی

- تغییر محتوای فعالیت در بخش `FrameLayout`

```
<androidx.fragment.app.FragmentContainerView
    android:id="@+id/nav_host_fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

```
android:name="androidx.navigation.fragment.NavHostFragment"
app:defaultNavHost="true"
app:navGraph="@navigation/nav_graph"
```

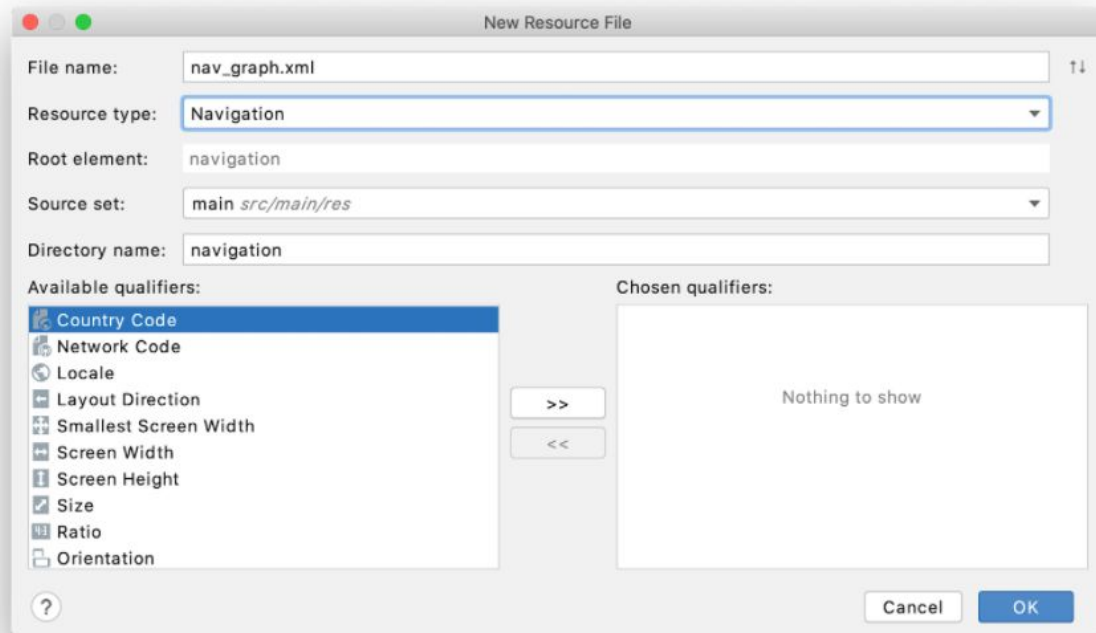
- اضافه کردن ویژگی‌های:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

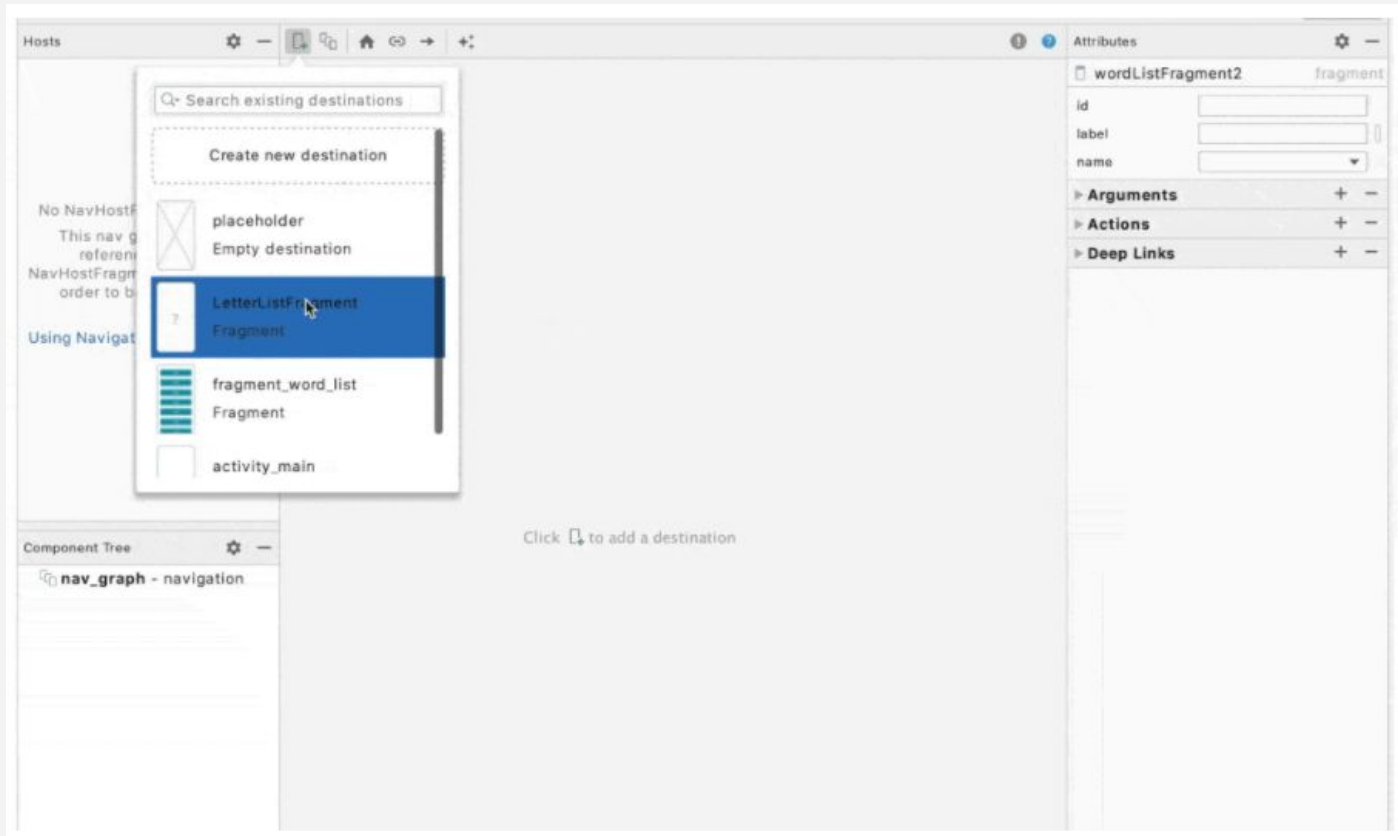
- اضافه کردن به `FrameLayout`:

ایجاد گراف ناوبری

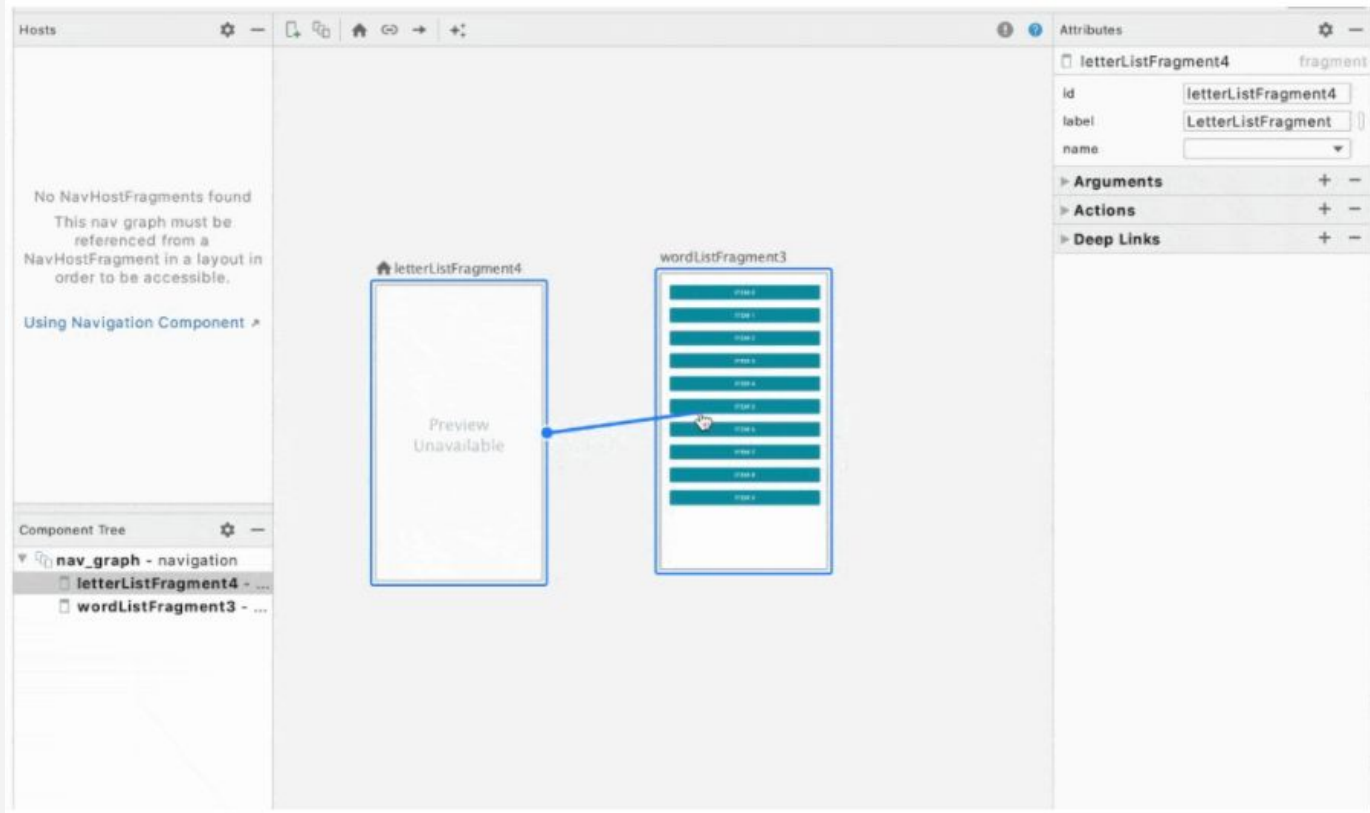
- اضافه کردن فایل XML در File > New > Android Resource File



اضافه کردن مقصد

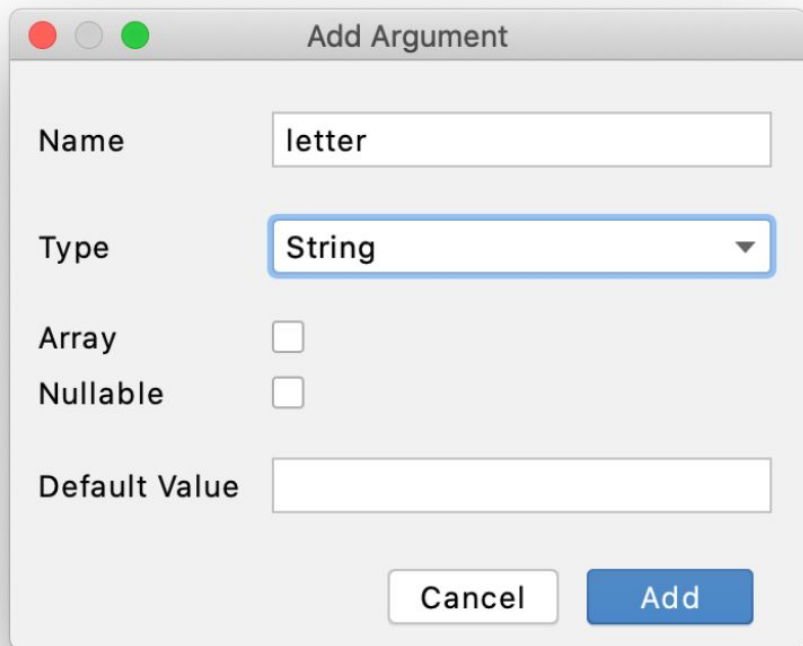


افزافه کردن عمل



انتقال اطلاعات بین قطعات

- تعریف یک آرگومان برای مقصد
 - انتخاب مقصد
 - اضافه کردن زیر Arguments



The image shows a 'Add Argument' dialog box with the following fields and controls:

- Name:** A text input field containing the value 'letter'.
- Type:** A dropdown menu with 'String' selected.
- Array:** A checkbox that is currently unchecked.
- Nullable:** A checkbox that is currently unchecked.
- Default Value:** An empty text input field.
- Buttons:** 'Cancel' and 'Add' buttons at the bottom right.

تعیین قطعه اولیه



Assign start destination

 letterListFragment

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

رفتن از یک قطعه به قطعه دیگر

- تغییر محتوای onClickListener در فایل LetterAdapter.kt

```
val action = LetterListFragmentDirections.actionLetterListFragmentToWordListFragment  
            (letter = holder.button.text.toString())
```

```
holder.view.findNavController().navigate(action)
```

به روز رسانی فعالیت اصلی

- تعریف ویژگی جدید در فعالیت

```
private lateinit var navController: NavController
```

- مقداردهی به آن در `onCreate`:

```
val navHostFragment = supportFragmentManager  
    .findFragmentById(R.id.nav_host_fragment) as NavHostFragment  
navController = navHostFragment.navController  
  
setupActionBarWithNavController(navController)
```

- پیاده سازی `onSupportNavigateUp`:

```
override fun onSupportNavigateUp(): Boolean {  
    return navController.navigateUp() || super.onSupportNavigateUp()  
}
```


استفاده از مقدار ارسال شده به قطعه

```
private lateinit var letterId: String
```

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    arguments?.let {  
        letterId = it.getString(LETTER).toString()  
    }  
}
```

```
recyclerView.adapter = WordAdapter(letterId, requireContext())
```

تعیین عنوان فعالیت برای هر قطعه

- تعیین مقدار label برای هر قطعه در گراف ناوبری
- اضافه کردن یک رشته جدید در فایل strings.xml

```
<string name="word_list_fragment_label">Words That Start With {letter}</string>
```

- تعیین مقدار label برای هر قطعه در گراف ناوبری

<input type="checkbox"/> letterListFragment	fragment
id	<input type="text" value="letterListFragment"/>
label	<input type="text" value="@string/app_name"/>
name	<input type="text"/>
► Arguments + -	

<input type="checkbox"/> wordListFragment	fragment
id	<input type="text" value="wordListFragment"/>
label	<input type="text" value="@string/word_list_fragment_label"/>
name	<input type="text"/>
▼ Arguments + -	



سوال؟