


☐

I'm not robot


reCAPTCHA

Continue

[illegible]

app on a split screen to see the navigation box. The reason why there is no layout with both the navigation box and the lower navigation is that the Guidelines of Material Design caution against this. Adding new destinations to NavigationView is easy. If you have a navigation box running with up and back navigation, you just need to add a new menu item. Open Menu/nav_drawer_menu.xml Add a new menu item for settings_dest<item android:id="@string/settings_dest" android:icon="@drawable/ic_settings" android:title="@string/settings_dest">now your navigation boxes show the settings screen as a destination. Good work! Navigation components also include deep link support. Deep links are a way to go into the middle of an app's navigation, whether it's from an actual URL link or a pending intent out of a notification. One of the advantages of using a navigation library to handle deep links is that it ensures that users start at the right destination with a corresponding stack back from other entry points, such as application widgets, notifications, or web links (covered in the next stage). Navigation provides the NavDeepLinkBuilder class to build a PendingIntent that to a specific destination. We will use NavDeepLinkBuilder to connect the app widget to its destination. Open DeepLinkAppWidgetProvider.kt Add PendingIntent built with NavDeepLinkBuilder: val args and Bundle () args.putString (myarg, From val PendingIntent - NavDeepLinkBuilder (context) .setGraph (R.id.deep_link_button R.navigation.mobile_navigation) .setDestination (R.id.deeplink_dest) .setArguments (args) .createPendingIntent () remoteViews.setOnClickPendingIntent setDestination determines where the link is going. setArguments includes any arguments you want to convey in your deep link. By default, NavDeepLinkBuilder will launch your launch. You can override this behavior by transmitting the action in context, or set a class of explicit action through setComponentName. Add the Deep Link widget to your home screen. Tap and hold on the home screen to see if you can add a widget. Click and holdscroll down to find the widget When you're done, you'll have a deep link widget. Click on the widget and make sure the Android destination opens with the right argument. He has to say: From the widget at the top because this is the argument you passed in DeepLinkLinkWidgetProvider. Make sure pressing the Back button takes you to home_dest destination. You can also call NavController createDeepLink to use NavController's context and current navigation schedule as a convenience. The back sheet for the deep link is defined by the navigational graph that you pass in. Backstack is generated using the directions in the app.startDestination. In this app we only have one action and one level of navigation, so a backstack will lead you to home_dest destination. More complex navigation can include nested navigation graphics. App.StartDestination at each level of nested graphs determines the backstack. For more information on deep links and nested graphs, check out the navigation principles. One of the most common uses of deep links is the ability of web links to open up activities in your app. Traditionally, you can use a intentional filter and link the URL to the action you want to open. The navigation library makes this extremely simple and allows you to map URLs directly to your destinations in your navigation schedule. Lt;deeplink- gt; It's an item that you can add to your destination on your schedule. Each item has one necessary attribute: app:uri. In addition to the live URI match, the following features are supported: URIs without a circuit are supposed to be http and https. For example, www.example.com will be matched by and . You can use seat holders as a to match one or more characters. The placeholder's line value is available in The Bundle arguments, which has a key with the same name. For example, will match the . You can use a wildcard. for zero or more characters. NavController will automatically process ACTION_VIEW intentions and look for appropriate deep links. At this point, you'll add a deep link to the www.example.com. Open mobile_navigation.xml Add the lt;deeplink element; deeplink_dest destination. Open AndroidManifest.xml Add a navigation tag schedule. deeplink_dest android:name/com.example.android.codelabs.navigation.DeepLinkFragment android:label="@string/deeplink tools:layout" android:defaultvalue/AndroidManifest.xml Add the lt;deeplink app:uri/www.example.com/z myarg/lt;deeplink> this will ensure that the appropriate filter of intent is generated by the MainActivity<filter> category android:name/android.intent.category.default category android:name/android.intent.category.mobile_navigation @navigation LAUNCHER In the project view, go to the app -> build -> Here you can take a look at the AndroidManifest created. Start the app at a deep link. There are two ways to do this: adb shell am start -n com.example.android.navigation/.MainActivity -a android.intent.action.VIEW -d www.example.com/urlTest in the search bar and a camouflage box will appear. Select Navigation codelab Open with the search bar (not in Chrome)Disambiguation dialogue Anyway, you should see the urlTest message on the screen. This was transmitted to a snippet from the URL. There's another part of the Codelab app that you can experiment with, and that's the basket button. This is a repetition of the skills that you learned during this codelab. This step doesn't include comments, so try it on your own: Create a new class of snippets Add a snippet as a destination for the navigation chart Use the basket icon to open your new piece class using NavigationUI to handle the menu. You are familiar with the basic concepts behind the navigation component! In this codelab you learned about: Navigation Count Structure NavHostFragment and NavController How to go to specific directions How to navigate in action How to convey the arguments between destinations, including using the new plug-in safeargs Navigation using menus, lower navigators, and navigation boxes Navigation Deep link You can continue to explore with this app or start using navigation in your own app. There's a lot more to try, including: Popping a destination from a backstack (or any backstack manipulation) nested navigation graphics Support for new destinations For more information on the navigation component, check out the documentation. If you're interested in exploring other architecture components, try the following codelabs: codelabs: android navigation architecture component tutorial. android navigation component tutorial java

normal_5f8c7ef63406b.pdf
normal_5f8a7567dee68.pdf
normal_5f8a4c3061b67.pdf
b44 bus schedule pdf
registers of scotland registration manual
11 2 probability and punnett squares
erosive lichen planus pdf
the successful match 2017 pdf
destiny 2 comp cheese
polynomials test review worksheet
cottonwood tree bark for sale
storyart apk full version
telecharger dictionnaire anglais francais apk
un coyote definition espagnol
calming guided sleep meditation
honda accord 2020 exl manual
doc to pdf offline
rocket royale mod apk 1.7.1
spotify premium ultimate apk latest pdf
galaxy buds manual download.pdf