

# Android with Kotlin Bootcamp

---

Day 2 - Building an Interactive Application



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](https://goobar.dev)



**Any questions from yesterday?**

What are we  
learning today?

**Activities & Fragments**

**Building User Interfaces**

**Resources**

**Building Interactive Screens**

**Building for a global audience**

**Navigating between screens**



**Let's check in on our “want to learns”**

Any Questions?

# Activities & Fragments

---



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](http://goobar.dev)

# Overview

- Understand what an Activity is**
- Understand the core Activity lifecycle**
- Understand the need for Fragments**
- Understand the core Fragment lifecycle**
- Understand common Activity/Fragment patterns**



# Activity

**An activity is a single, focused thing that the user can do**

<https://developer.android.com/reference/android/app/Activity>

# Activity

**1 of 4 core Android components**

**Must be declared in `AndroidManifest.xml`**

**Common app entry point**

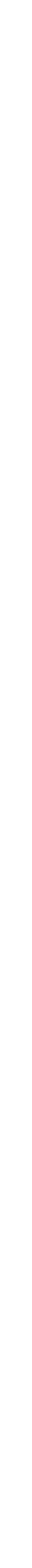


**Application starts**

**New process**

**New task**

**Launcher Activity created**



On Application Start

Activity

**How to start an Activity?**

**How to start an Activity for a result?**

**How to finish an Activity?**

# Intent

**An intent is an abstract description of an operation to be performed**

<https://developer.android.com/reference/android/content/Intent>

# Activity Lifecycle

**The set of States and Transitions an Activity may go through during normal operation of an Android application.**

[https://developer.android.com/guide/components/images/activity\\_lifecycle.png](https://developer.android.com/guide/components/images/activity_lifecycle.png)

# What is a Fragment?

# Fragment

**A Fragment represents a reusable portion of your app's UI.**

<https://developer.android.com/guide/fragments>

# Fragment

**Introduced with Android 3.0**

**Manages own layout, lifecycle, input**

**Modular & reusable**

# Activity vs Fragment

<b>Activity</b>	<b>Fragment</b>
<b>Core Android component</b>	<b>Add on to core framework</b>
<b>Required for application UI</b>	<b>Not required</b>
<b>May contain 0, 1, or many fragments</b>	<b>Must be contained within Activity</b>
<b>Has a unique lifecycle</b>	<b>Can be reused across Activities</b>
	<b>May contain child Fragments</b>
	<b>Has a unique lifecycle</b>

# Fragment Key Terms

**Fragment  
Manager**

**Fragment  
Transaction**

**Fragment  
Backstack**

# Fragment Lifecycle

**The set of States and Transitions a Fragment may go through during normal application use.**

<https://developer.android.com/guide/fragments/lifecycle>

Fragment

**How to create a Fragment?**

**How to create Fragment UI?**

**How show a Fragment?**

# Building User Interfaces

---



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](https://goobar.dev)

# Overview

**Discover commonly used Views**

**Explore commonly used ViewGroups**

**Understand how to use XML and UI Designer**

**Understand how to use ConstraintLayout**

# Android UI Toolkits



**XML**



**Jetpack Compose**

# XML to the Screen

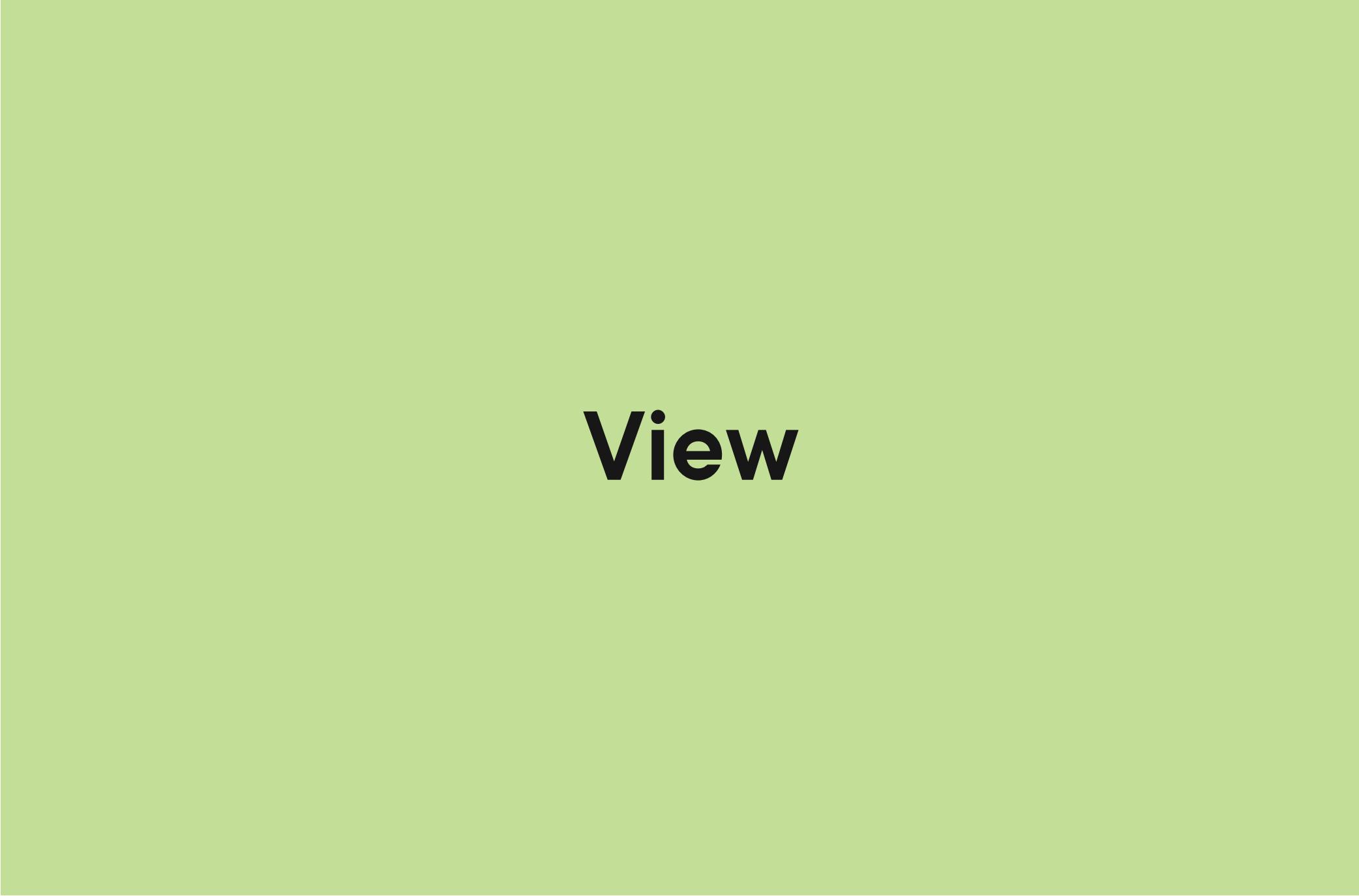
**Define layout in XML**

**Set into Activity/Fragment**

**Inflated into View**

**Interact via code**

# Views & ViewGroups



**View**



**ViewGroup**

# Common ViewGroups

**FrameLayout**

**LinearLayout**

**RelativeLayout**

**ConstraintLayout**

# Common Views

**TextView**

**Button**

**FloatingActionButton**

**Toolbar**

**ImageView**

**EditText**

# Collection-Based Views

**Spinner**

**ListView**

**RecyclerView**

**ViewPager2**

# ConstraintLayout



- Relative positioning**
- Margins**
- Centering**
- Dimensions**
- Chains**
- Virtual helpers**

Toolbar

Menus

Navigation icon

Title

Examples

# BottomNavigation

**Display 2-5 tabs along bottom of screen**

**Populate options via Menu**

**Display Fragment when items are clicked**

**Examples**

# TextInputLayout

**Provide more interesting edit experience**

**Hint & Error states**

**Wraps an EditText**

**Examples**

# Working with Resources

---



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](https://goobar.dev)

# Overview

- Understand the Android resource system**
- Learn to create configuration-based resources**
- Understand how to work with String resources**
- Understand how to work with Drawables**
- Learn about other Android resource types**

# Android Resources

**The Android resource system keeps track of all non-code assets associated with an application.**

<https://developer.android.com/guide/topics/resources/available-resources>

# Android Resources

**Layouts**

**Menus**

**Styles/Themes**

**Strings**

**Drawables**

**Values**

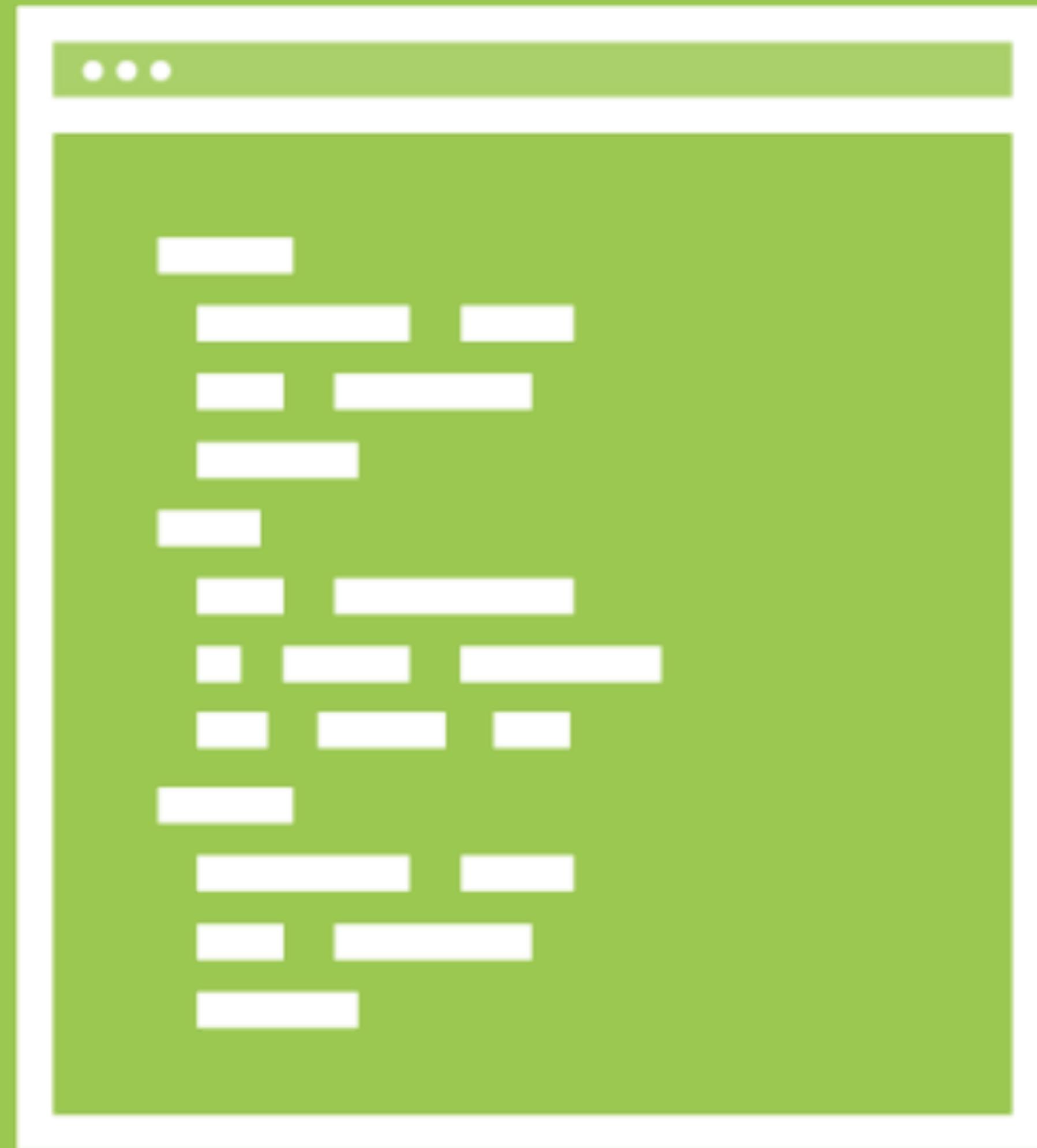
# Resources

**How to create a new Resource?**

**How to reference a Resource from code?**

**How to reference a Resource from XML?**

**How to view all Resources?**



# Resource Qualifiers

Almost every app should provide alternative resources to support specific device configurations.

# Resource Qualifiers

**What is a qualifier?**

**What is a Configuration?**

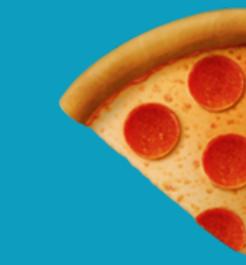
**How to query these?**

**Examples**

# Context

**Interface to global information about an application environment ... It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.**

<https://developer.android.com/reference/android/content/Context>



Lunch

# Building Dynamic Screens

---



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](https://goobar.dev)

# Overview

**Understand state restoration**

**Explore Toasts and Snackbars**

**Understand common input validation patterns**

**Understand how to show/hide Views**



# Handling Configuration Changes

**Configuration Change**

**SaveInstanceState**

**Recreation**

**RestoreInstanceState**

When is state  
saved?

**Does your view have an id?**  
**Have users entered input?**  
**Do you want to preserve context?**

# Testing State Restoration

**Rotation**

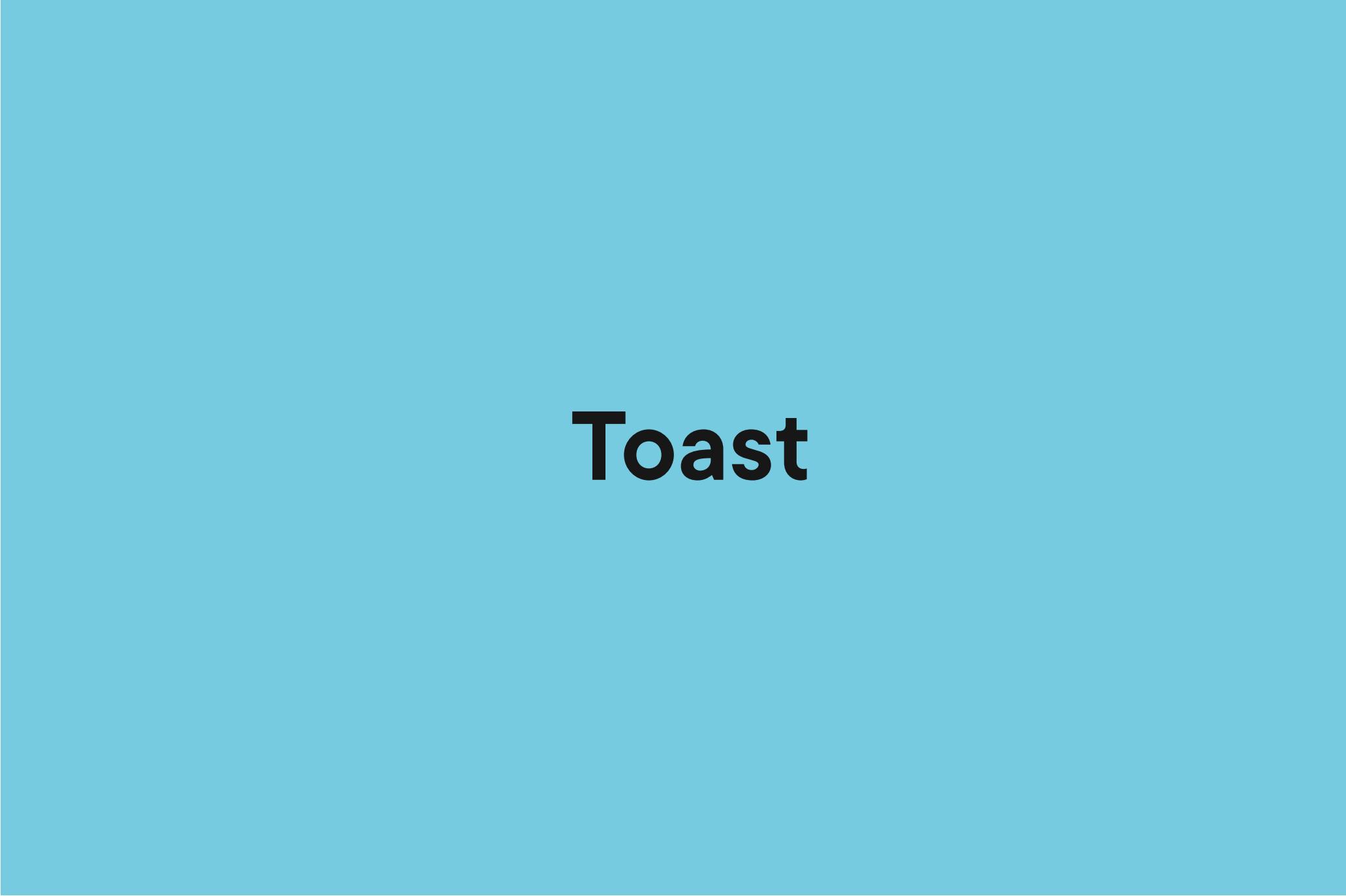
**Dismiss app from Task Stack**

**Enable ‘Don’t Keep Activities’**

**Kill via adb**

How to display feedback  
to a user?

# Simple User Feedback



**Toast**



**Snackbar**

# Building for a Global Audience

---



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](https://goobar.dev)

# Overview

**Better understand Android fragmentation**  
**Understand the challenges of building for a global audience**  
**Learn about responsive design**  
**Better understand String and Date localization**

What is the major challenge in  
Android development today?

# Fragmentation is a major challenge

[https://www.opensignal.com/sites/opensignal-com/files/data/reports/  
global/data-2015-08/2015\\_08\\_fragmentation\\_report.pdf](https://www.opensignal.com/sites/opensignal-com/files/data/reports/global/data-2015-08/2015_08_fragmentation_report.pdf)

# How to choose supported users?

<https://gs.statcounter.com/os-version-market-share/android/mobile-tablet/worldwide>

# Challenges of Android Development

**Manufacturers**

**OS Variants**

**Form Factors**

**API Versions**

**Languages**

**Hardware**

Manufacturer /  
Variants

- Platform-specific apis?
- Platform-specific SDKs?
- Platform-specific marketplaces?
- Platform-specific bugs?

# Form Factors

**Phone**

**Tablet**

**Chromebooks**

**Foldables**

**Watch**

**Auto**

# Form Factors

- Extra small screen?**
- Extra large screen?**
- Screen density?**
- Folding?**
- Resizable display?**
- Multiple displays?**

# Api Versions

- Permissions differences?
- Theming differences?
- Added/missing apis?
- Build implications?

# Localization



- Are you using String resources?**
- Are your Strings translated?**
- How often do you update Strings?**
- Does your UI adapt to long Strings?**
- Does your UI support RTL languages?**
- How do you test different locales?**
- How do you format date/time?**

# Hardware

**Screen Density**

**Screen Size**

**Keyboard**

**Camera(s)**

**Accelerometer**

# Responsive Design

**Designing, and building, applications that adapt to whatever form factors, hardwares, or other configurations on which they're run.**

# Responsive Design

**Support Resource Qualifiers**

**Design for Common Form Factors**

**Test Across Locales**

**Leverage Compat APIs**

**Start from the Beginning**

# Navigating Between Screens

---



**Nate Ebel**

Android Developer & Instructor

@n8ebel [goobar.dev](https://goobar.dev)

# Overview

**Understand challenges of in-app navigation**

**Understand interface-based navigation**

**Understand how to use Navigation Component**

# Challenges of In-App Navigation

**Separation of Concerns**

**Modularization**

**Deeplinking**

**Complexity**

# Types of Navigation

**Manual**

**Indirect**

**Navigation  
Component**

# Android Jetpack Navigation Component

**Android Jetpack's Navigation component helps you implement navigation, from simple button clicks to more complex patterns, such as app bars and the navigation drawer.**

<https://developer.android.com/guide/navigation>

# Navigation Component

- Visual navigation graph**
- Strongly typed navigation actions**
- Built-in deeplink support**
- Integration with other UI components**
- Simplified animations**

# NavHost

**A host is a single context or container for navigation via a NavController**

<https://developer.android.com/reference/androidx/navigation/NavHost>

# NavController

**NavController manages app navigation within a NavHost**

<https://developer.android.com/reference/androidx/navigation NavController>

# Navigation Graph

**A resource file that contains all of your destinations and actions. The graph represents all of your app's navigation paths.**

<https://developer.android.com/guide/navigation/navigation-getting-started>

# Destination

**Different content areas in your app. Typically a Fragment.**

<https://developer.android.com/guide/navigation/navigation-getting-started>

# Action

**A logical connections between your destinations that represent paths that users can take.**

<https://developer.android.com/guide/navigation/navigation-getting-started>



# Office Hours