



30-Day Android Challenge

From Basics to Advanced

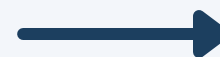
Day 1

(A Comprehensive Journey into Activity and Lifecycle) 🚀



Esmaeeil Moradi

@esmaeeil-moradi700



Day 1

(A Comprehensive Journey into Activity and Lifecycle) 🚀

Hello to all Android enthusiasts and experts! 🙌

Today marks the first step in our 30-day challenge, and we're diving deep into one of the most fundamental yet intricate concepts in Android development: Activity and its Lifecycle. Whether you're a curious beginner, an experienced developer, or even someone keen to understand the inner workings of AOSP, a thorough grasp of this topic is crucial for building flawless applications.



Esmaeel Moradi

@esmaeel-moradi700



What is an Activity?

(Basic and Expert Perspectives)

- **Basic:** Simply put, an Activity is like an interactive screen in your application that the user can engage with. Everything you see on your phone's display, from buttons to lists, is displayed within an Activity.
- **Expert:** At the Framework level, an Activity is a key component in Android's application architecture, responsible for managing the application's Context, resources, Lifecycle Callbacks, and communication with the operating system through the ActivityManagerService (AMS). An Activity essentially serves as the coordination point between incoming Intents, the user interface, and the application's state management.



Esmaeel Moradi

@esmaeel-moradi700



The Activity Lifecycle: A Journey from Birth to Death (with a Deep Dive into AOSP)

Every Activity goes through a series of stages during its lifespan, with the ActivityManagerService (AMS) playing a pivotal role in managing them at the operating system level:

- **onCreate() (Birth):** The system creates an instance of the Activity. This is where UI initialization and variable setup occur. At the AOSP level, the AMS establishes a Window Session for this Activity, which acts as a channel for communication with the WindowManagerService (WMS), and initial resources are allocated.



Esmaeel Moradi
@esmaeel-moradi700



- **onStart() (Start):** The Activity becomes visible to the user. It's added to the WMS, and the Measure and Layout process for determining the size and position of Views begins.
- **onResume() (Active):** The Activity comes into the Foreground and receives input focus. The AMS continuously manages the state of Activities and decides which one is in the Resumed state.
- **onPause() (Temporary Pause):** The Activity is no longer in the Foreground and might not be fully visible. This is a critical opportunity to save important data and release temporary resources. In this stage, the Activity loses input focus.



Esmaeel Moradi

@esmaeel-moradi700

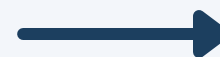


- **onStop() (Complete Stop):** The Activity is no longer visible to the user and is in the Background. UI processes are halted, and UI resources might be released. The system might destroy it to reclaim memory.
- **onRestart() (Return):** Called when a previously stopped Activity is being started again.
- **onDestroy() (Death):** Called before the Activity is completely removed from memory. This is where you should release all remaining resources. Be aware that in low-memory situations, this method might not be called. The AMS disconnects the Activity's communication with the WMS.



Esmaeel Moradi

@esmaeel-moradi700



Key Insights and Advanced Scenarios:

- **State Management: Utilizing ViewModel to persist data across configuration changes instead of solely relying on onSaveInstanceState().**
- **Importance of onPause(): Crucial for saving volatile data and releasing resources to prevent slow app switching.**
- **Process Death: Understanding that the system might terminate background Activities without prior notice and planning for state preservation.**



Esmaeel Moradi

@esmaeel-moradi700



Expert Best Practices:

- **Avoid performing heavy operations on the Main Thread during Lifecycle Callbacks.**
- **Leverage Architecture Components like ViewModel and LiveData for lifecycle-aware state management.**
- **Be mindful of Lifecycle Owners and associated Scopes for managing Coroutines.**



Esmaeel Moradi
@esmaeeil-moradi700



? Interview Questions:

- **What is Activity?**

Answer: An Activity is a fundamental component in Android that provides an interactive screen for the user and is responsible for managing the UI lifecycle, application state, and communication with the operating system via the AMS.

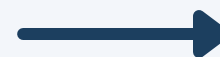
- **Describe Activity lifecycle.**

Answer: The Activity lifecycle comprises the methods `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onRestart()`, and `onDestroy()`, which are invoked by the operating system in response to changes in the Activity's state, each serving specific roles in managing resources and application state.



Esmaeel Moradi

@esmaeel-moradi700



? Challenge Question for Experts (with AOSP perspective): Considering the interaction

- **between Activity, AMS, and WMS, what performance and architectural considerations are crucial for managing the Activity lifecycle in high-performance and large-scale applications?**
- **We look forward to your valuable insights and experiences! Let's elevate the level of Android development together by sharing our knowledge. 🚀**



Esmaeel Moradi
@esmaeeil-moradi700

We look forward to your
valuable insights and
experiences! Let's elevate
the level of Android
development together by
sharing our knowledge.



LEAVE A COMMENT BELOW



Esmaeel Moradi

@esmaeel-moradi700