Android Activity Lifecycle

Qiong LIU

$21 \ {\rm octobre} \ 2024$

1 Activity Lifecycle



FIGURE 1 – Android Activity Lifecycle Diagram

This diagram represents the Android Activity Lifecycle, showing how an Activity transitions between different states based on user interactions or system constraints. Let's break it down step by step :

1. Activity Launched The Activity starts when it is first launched. This typically happens when the user opens an app or navigates to a specific screen.

- 2. **onCreate()** This is the first callback method invoked when the Activity is created. It's used to initialize the Activity, set up the user interface (via **setContentView()**), and restore any saved instance state.
- 3. **onStart()** After **onCreate()**, the Activity enters the **Started** state, and **onStart()** is called. At this point, the Activity becomes visible to the user but is not yet in the foreground for interaction.
- 4. **onResume()** The **onResume()** callback brings the Activity to the foreground, making it active and interactable by the user. The Activity is now in the **Resumed (Running)** state.
- 5. Activity Running In this state, the Activity is fully in the foreground and is responding to user input. The Activity stays in this state as long as it is the focus of user interaction.
- 6. **onPause()** If another activity comes into the foreground (e.g., a phone call or a new app screen is opened), **onPause()** is triggered. The Activity is now in the **Paused** state. It is still visible (or partly visible), but it cannot interact with the user. The Activity remains in this state if the new activity is partly transparent or does not cover the entire screen.
- 7. **onStop()** If the Activity is completely hidden (i.e., the new activity fully covers the screen), **onStop()** is called, and the Activity moves to the **Stopped** state. The Activity is no longer visible, but its state and instance are still retained in memory. If resources are needed by higher-priority apps, the system may kill the stopped Activity.
- 8. onDestroy() When the Activity is finishing (e.g., the user presses the back button or the Activity is explicitly closed), onDestroy() is called, and the Activity is completely shut down. In this state, all resources held by the Activity are cleaned up.
- 9. App Process Killed If the system requires memory for other higher-priority processes, the Activity may be killed even if it's in the paused or stopped state. If the Activity needs to be restarted later, it will start from onCreate() again unless it was in the background and can be restored.
- 10. **onRestart()** If the Activity is paused or stopped and the user navigates back to it (e.g., by pressing the back button or selecting it from the recent apps list), **onRestart()** is called. This callback signals that the Activity is being reactivated, and after this, **onStart()** and **onResume()** are triggered to bring the Activity back to the foreground.