Android Activity Lifecycle

Prashanth B S[1]

[1]Department of Information Science & Engineering, Nitte Meenakshi Institute of Technology,
Yelahanka - 560064, Bengaluru

1 Android Activity Life-cycle

As a user navigates through, out of, and back to your app, the Activity instances in your app transition through different states in their lifecycle. The Activity class provides a number of callbacks that allow the activity to know that a state has changed: that the system is creating, stopping, or resuming an activity, or destroying the process in which the activity resides[1].Within the lifecycle callback methods, you can declare how your activity behaves when the user leaves and re-enters the activity.

2 Activity-lifecycle callbacks

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

1. *onCreate*()- You must implement this callback, which fires when the system first creates the activity. On activity creation, the activity enters the Created state. In the onCreate() method, you perform basic application startup logic that should happen only once for the entire life of the activity.

2. *onStart*()- When the activity enters the Started state, the system invokes this callback. The onStart() call makes the activity visible to the user, as the app prepares for the activity to enter the foreground and become interactive.

3. *onResume*()- When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the onResume() callback. This is the state in which the app interacts with the user.

4. *onPause*()- The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed); it indicates that the activity is no longer in the foreground (though it may still be visible if the user is in multi-window mode)

5. *onStop*()-When your activity is no longer visible to the user, it has entered the Stopped state, and the system invokes the onStop() callback. This may occur, for example, when a newly launched activity covers the entire screen. The system may also call onStop() when the activity has finished running, and is about to be terminated.

6. *onDestroy*()-onDestroy() is called before the activity is destroyed.

The overall process of using these callbacks are demonstrate in the figure 1,

[1]https://developer.android.com/guide/components/activities/activity-lifecycle

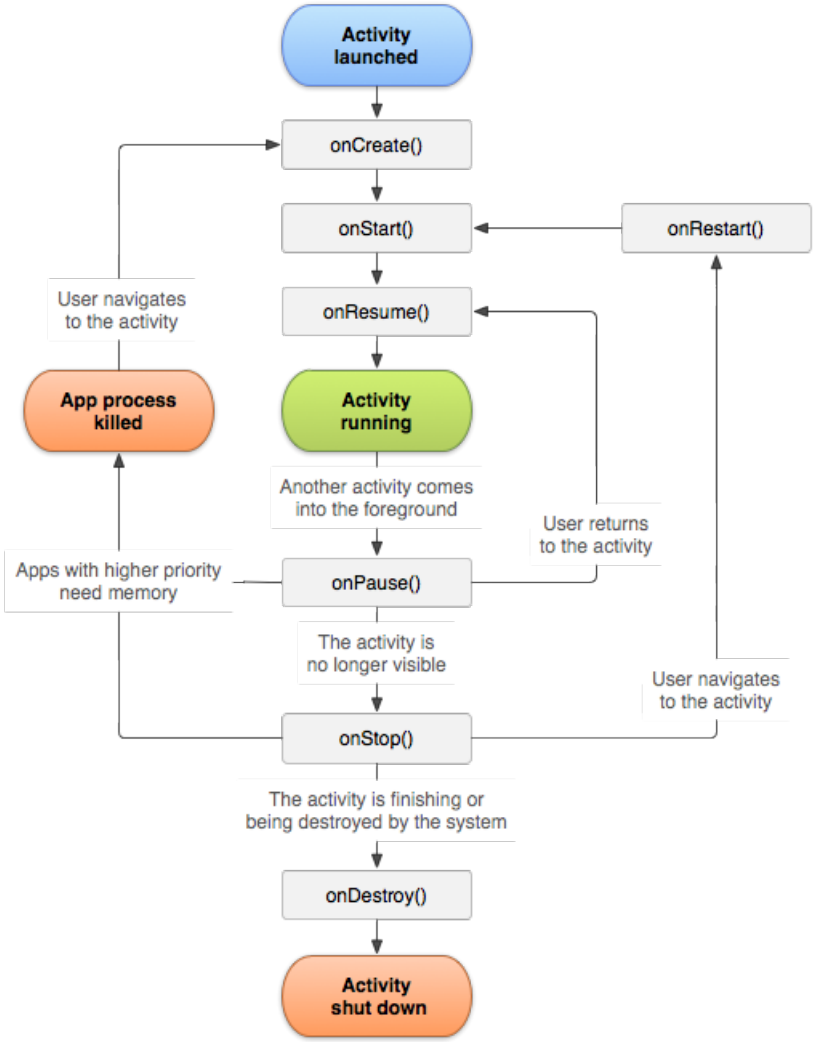


Figure 1: Android Activity Lifecycle

3 Exercise-I

Create an Android App to demonstrate the Working of Android Activity LifeCycle. The following are the steps,

- Create an Empty Project, By default, onCreate() function will be created, Add a Toast message after the super keyword in onCreate as shown below,

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(this, "ACTIVITY CREATED", Toast.LENGTH_SHORT).show();
}
```

- Press CTRL+O after the oncreate() Function, A menu will pop up. Choose the override methods such as onStart(), onResume(), onPause(), onStop(), and onDestroy() resepectively as shown in figure 2 ,

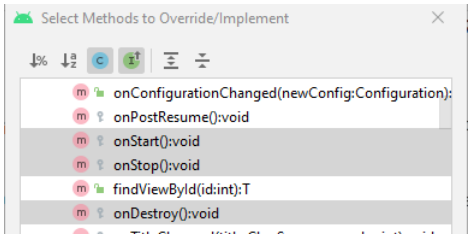


Figure 2: Selecting Override functions

- Add Toast messages in the respective callbacks such as onStart(), onResume(), onPause(), onStop(), and onDestroy() as shown below for onStart() and onPause().

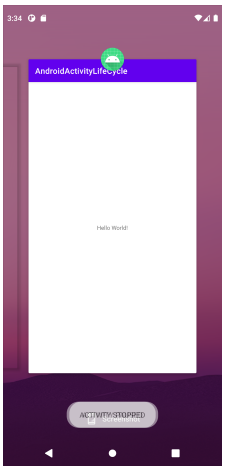
```
@Override
protected void onStart() {
        super.onStart();
        Toast.makeText(this, "ACTIVITY STARTED", Toast.LENGTH_SHORT).show();
}

@Override
protected void onPause() {
        super.onPause();
        Toast.makeText(this, "ACTIVITY PAUSED", Toast.LENGTH_SHORT).show();
}
```

- Similarly, Implement the Toast messages for onResume(), onStop(), and onDestroy() respectively.

4 Output

Sample output window of the app is as shown below figure 3,



(a) Activity Paused

(b) Activity Resumed

Figure 3: Activity CALLBACKS