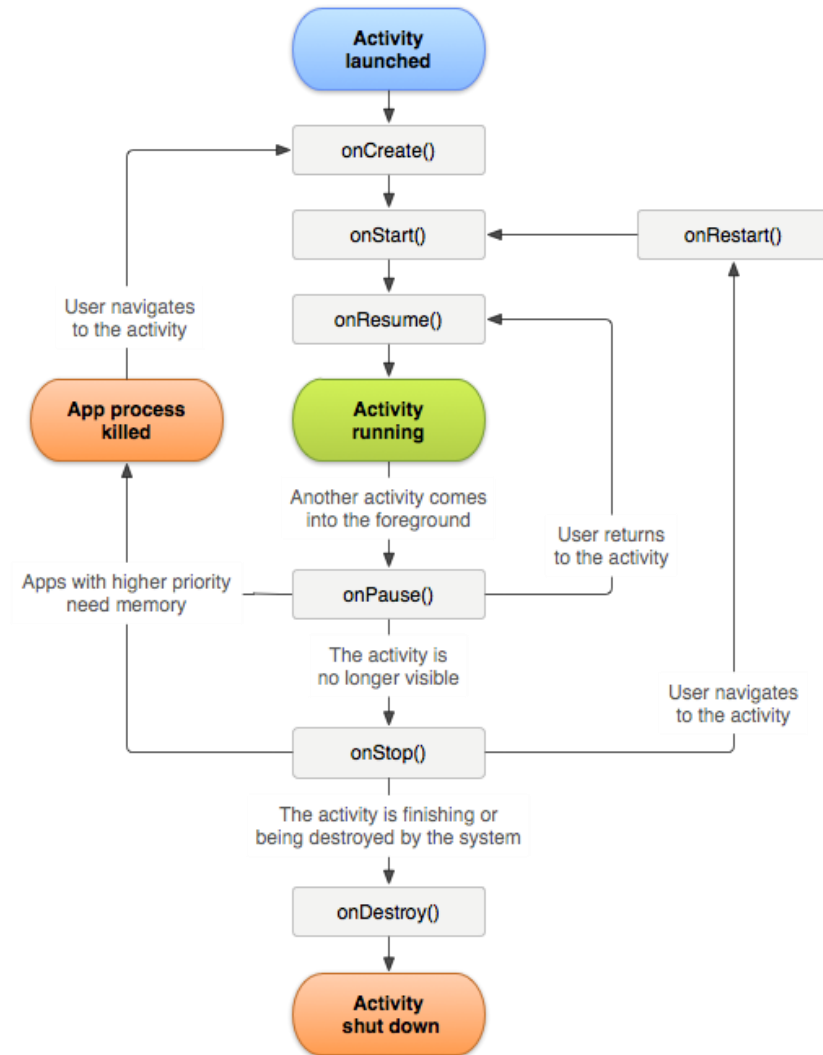


Touching On States: An In-depth Analysis

Al-Kandari, AbdulMuhsin

Blessing, James

Quick Overview: What We Know?



Placement: Where Do They Go?

```
public class myActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
    }  
  
    @Override  
    protected void onPause() {  
        super.onPause();  
    }  
  
    @Override  
    protected void onRestart() {  
        super.onRestart();  
    }  
  
    @Override  
    protected void onResume() {  
        super.onResume();  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
    }  
  
    protected void onStop() {  
        super.onStop();  
    }  
}
```

Dynamic: How to Get Her There?

`onCreate()`, `onStart()`, and `onDestroy()`

- ▶ `onCreate()`: Called when the activity is first created, it takes a `Bundle` object as its parameter which contains the activity's previously frozen state if one exists. Where you should do all of your setup: create views, bind data to lists, etc. Always followed by `onStart()`.
- ▶ `onStart()`: Called when the activity is becoming visible to the user. Followed by `onResume()` if the activity comes to the foreground, or `onStop()` if it becomes hidden. (`onRestart()` is a closely related method to `onStart`, it is called after your activity has been stopped, coding this method is optional.)
- ▶ `onDestroy()`: The final call before an activity is destroyed. This happens either because the activity is finishing (`finish()` was called) or the android system is destroying the activity to save space. Distinguishing between these two scenarios is done with `isFinishing()`.

Static: You're Doing Your Job Right onResume()

- ▶ Called when the activity will start interacting with the user.
- ▶ When this fires the activity is top of the activity stack.
- ▶ It signals the only state in which the user can input into the activity.
- ▶ Always followed by onPause();
- ▶ The best time to make sure that the needed resources are activated.

Static: Need A Break?

onPause()

- ▶ Called when the system is about to start resuming a previous activity or stopping the current activity
- ▶ Typically used to commit unsaved changes, stop animations and stop other CPU consuming actions.
- ▶ This method should be very quick, no CPU intensive actions, since the next activity will not Resume until the system returns from this method.
- ▶ It is followed by onResume() if the activity returns back to the front, or by onStop() if it becomes invisible to the user.

Static: Satisfied onStop()

- ▶ Called once the activity is no longer visible.
- ▶ This may happen because a new activity is being started, an existing one is being brought in front of this one, or this one is being destroyed.
- ▶ It is followed by either `onRestart()` if the activity is coming back to interact with the user, or `onDestroy()` if this activity is going away.
- ▶ The app is destructible at this point.