
globsyn
Taking People To The Next Level

globsyn

globsyn
finishing school

Services

Services

- A service is an application component that can run in the background to perform work even while the user is in a different application
- It allows other components to be bound to it, to interact with it and perform interprocess communication.
- A service can essentially take two forms:
  - **Started:** A service is "started" when an application component starts it by calling startService().
- Once started, a service runs in the background irrespective of the component that started it.
- A started service performs a single operation and does not return a result to the caller and stops once the service is done.

Services

- **Bound:** when an application component binds a service to it by calling bindService() a bound takes place.

- A bound service offers a client-server interface that allows components to interact with the service.
- A bound service runs only as long as another application component is bound to it.
- Multiple components can bind to the service at once, but when all of them unbind, the service is destroyed.
- A service runs in the main thread of the application that hosts it, by default.

Services

- To create a service, a subclass of Service is created.
- To implement, some callback methods are necessary, these include the following:

OnStartCommand(): The system calls this method when another component, requests that the service be started, by callingstartService().

Once the started service is no longer needed it can be stopped by calling stopSelf() orstopService().

onBind()The system calls this method when another component wants to bind with the service

.onCreate: system calls this method when the service is first created, to perform one-time setup procedures

Services

onDestroy(): The system calls this method when the service is no longer used and is being destroyed.

- This is the last call the service receives.
- Like activities, all services must be declared in the application's manifest file.
- To declare your service, add a <service> element as a child of the <application>element.
- A started service is one that another component starts by calling startService() resulting in a call to the service's onStartCommand() method.

Services

- A service can be started from an activity or other application component by passing an Intent to startService().
- The Android system calls the onStartCommand() method and passes it to the intent.
- The service must stop itself by calling stopSelf() or another component can stop it by calling stopService().
- To create a bound service, the interface has to be defined to specify how a client can communicate with the service.

Services

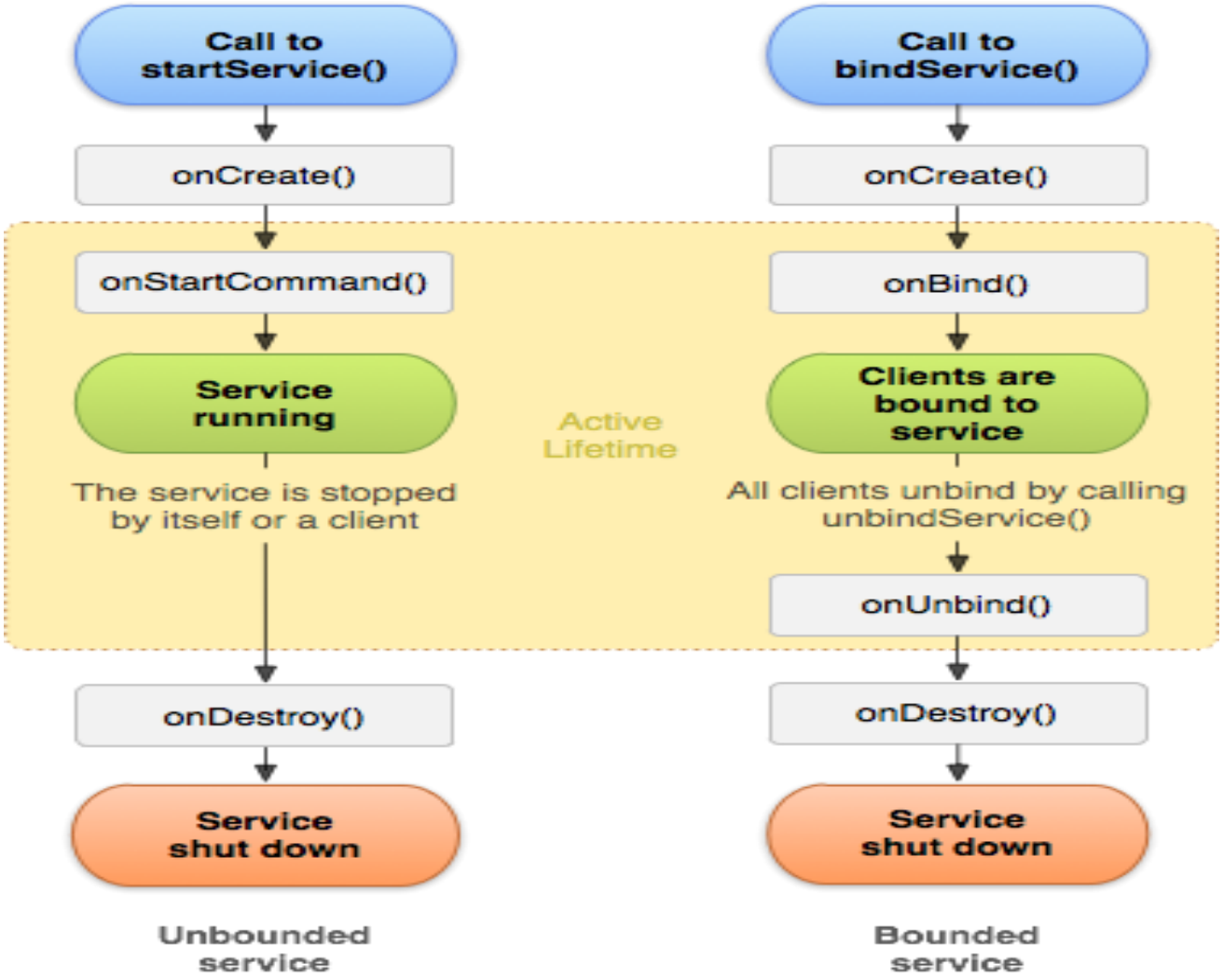
- This interface between the service and a client must be an implementation of Ibinder,
- Once the client receives the Ibinder service, it can begin interacting with the service through that interface.
- A service is required to notify the user of events using Toast Notifications or Status Bar Notifications
- A toast notification is a message that appears on the surface of the current window for a moment then disappears,
- A status bar notification provides an icon in the status bar with a message, for the user to select and take an action.

Services

- Similar to an activity lifecycle the services also have an active lifecycle.
- There are two ways in describing the lifecycle:
- One is when an activity calls for a service it is first created.
- The created service is then made to run.
- When the service is no longer needed the service is stopped and destroyed.
- Another is when a request to bound emerges, the service is bound to other components.

Services

- The interface are specifies and the communication is performed.
- Once this completes the unbinding process takes place and the services are destroyed
- This is explained with the help of a flow chart -


Call to startService()
onCreate()
onStartCommand()
Service running
The service is stopped by itself or a client
onDestroy()
Service shut down
Unbounded service
Active Lifetime
Call to bindService()
onCreate()
onBind()
Clients are bound to service
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
Service shut down
Bounded service

Services

- AIDL (Android Interface Definition Language) defines the programming interface that both the client and service agree upon in order to communicate with each other using interprocess communication (IPC).
- On Android, one process cannot normally access the memory of another process.
- So to talk, they decompose their objects into primitives that the operating system can understand, and send these objects across that boundary.
- The code to send across the boundaries is tedious to write, so Android handles it with AIDL.

Taking People To The Next Level ...