## What are Android Services?

Android Services are the application components that run in the background. We can understand it as a process that doesn’t need any direct user interaction. As they perform long-running processes without user intervention, they have no User Interface. They can be connected to other components and do ****inter-process communication (IPC)****.

### Types of Android Services

When we talk about services, they can be of three types:

#### 1. Foreground Services

Foreground services are those services that are visible to the users. The users can interact with them at ease and track what’s happening. These services continue to run even when users are using other applications.

*The perfect example of this is Music Player and Downloading.*

#### 2. Background Services

These services run in the background, such that the user can’t see or access them. These are the tasks that don’t need the user to know them.

*Syncing and Storing data can be the best example.*

#### 3. Bound Services

Bound service runs as long as some other application componentis bound to it. Many components can bind to one service at a time, but once they all unbind, the service will destroy.

To bind an application component to the service, ****bindService()**** is used.

**Lifecycle of Service in Android**

1. \*\*Created\*\*:

When a service is first created using `onCreate()`. At this stage, the service is initialized, but it hasn't started running yet.

1. \*\*Started\*\*:

When a service is started using `startService()` method, it enters the started state. The service remains in this state until it stops itself or is stopped by calling `stopService()` or `stopSelf()`.

1. \*\*Bound\*\*:

A service can also be bound to other application components such as activities. When a component binds to the service using `bindService()`, the service enters the bound state. In this state, the service provides an interface for the client to interact with it. When all clients unbind from the service using `unbindService()`, the service is destroyed unless it was also started with `startService()`.

1. \*\*Running\*\*:

A service is in the running state when it's performing its work. It stays in this state until it's stopped or destroyed.

1. \*\*Destroyed\*\*:

When a service is no longer needed and is stopped using `stopSelf()` or `stopService()`, or when the system destroys it to free up resources, it enters the destroyed state. At this point, the service is removed from memory, and all resources associated with it are released.