



Service의 생명주기

Service?

Service의 생명주기

1. onCreate()
2. onStartCommand(Intent, int, int)
3. onBind(Intent)
4. onUnbind(Intent)
5. onRebind
6. onDestroy()

Service의 형태

Started Service

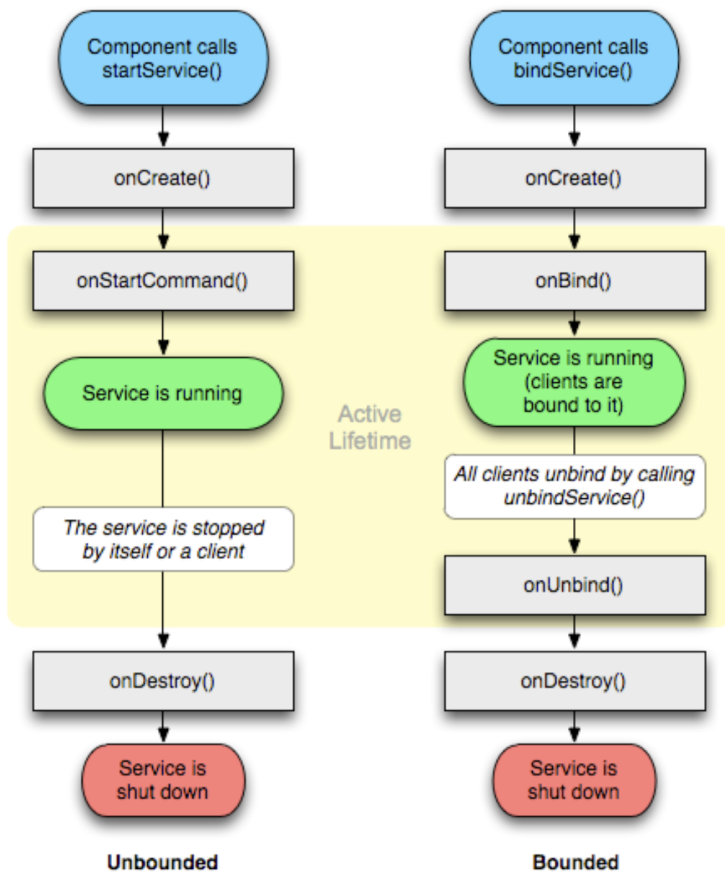
Bound Service

생명주기 흐름

Service?

- 안드로이드 4대 컴포넌트 중 하나
- Activity가 종료되어 있는 상태에서도 동작하기 위해서 만들어진 컴포넌트
- Background에서 동작
- UI를 가지지 않음

Service의 생명주기



Service 생명주기

1. onCreate()

서비스 생성(Created)

- Service가 처음으로 생성될 때 호출
- 서비스 초기화에 필요한 모든 작업 수행
- 생명주기동안 한 번만 호출 됨

2. onStartCommand(Intent, int, int)

서비스 시작(Started)

- Service가 시작될 때 호출
- 각각의 호출에 대한 처리를 정의하며, 시작 유형과 관련된 작업 수행
- 백그라운드 작업이나 다른 구성 요소와의 통신 처리

- 생명주기 동안 여러번 호출 가능

3. onBind(Intent)

서비스 바인딩(Bound)

- 다른 구성 요소가 Service에 바인딩 하려고 할 때 호출
- 다른 컴포넌트가 Service에 접근하고 통신 가능
- IBinder를 반환해야 하며 IBinder는 서비스와 클라이언트 간의 인터페이스 제공
 - null을 반환하기도 하는데, 이는 클라이언트가 서버와 바인드할 수 없음을 의미

4. onUnbind(Intent)

서비스 언바인딩(Unbound)

- unbindService()에 의해 서비스 바인딩이 해제될 때 호출
- 모든 클라이언트가 서비스에서 언바인드 되었을 때 호출
- 필요에 따라 재정의 할 수 있음
- 선택적임

5. onRebind

서비스 재바인딩(Rebound)

- 이미 바인딩 된 상태에서 다시 바인딩 될 때 호출
- onUnbind() 이후에 새 클라이언트가 서비스에 다시 바인딩하려고 할 때 호출
- 선택적임

6. onDestroy()

서비스 종료(Destroyed)

- 서비스가 소멸될 때 호출
- 리소스 정리 및 마무리 작업 수행

Service의 형태

Started Service

▼ 시작 및 정지

- `startService()` 메소드로 시작
- `stopSelf()` 또는 `stopService()` 메소드를 호출하여 종료

▼ 생명주기

`onCreate()`

- 서비스가 처음 생성될 때 호출

`onStartCommand()`

- `startService()` 메소드로 서비스가 시작될 때 호출

`onDestroy()`

- 서비스가 종료될 때 호출

Bound Service

▼ 바인딩과 해제

- `bindService()` 메소드를 사용하여 다른 컴포넌트에 바인딩
- 바인딩된 서비스는 바인딩을 사용하는 클라이언트와 상호작용 가능
- `unbindService()` 메소드로 바인딩 해제

▼ 생명주기

`onCreate()`

- 서비스가 처음 생성될 때 호출

`onBind()`

- `bindService()` 메소드로 서비스에 처음 바인딩될 때 호출

`onUnbind()`

- 모든 클라이언트가 바인딩을 해제할 때 호출

`onDestroy()`

- 서비스가 종료될 때 호출

생명주기 흐름

Started Service

- onCreate() → onStartCommand() (실행 중) → stopSelf() 또는 stopService() → onDestroy()

Bound Service

- onCreate() → onBind() (바인딩 중) → onUnbind() (모든 클라이언트가 바인딩 해제 시) → onDestroy()