

Android 주기적인 조회 서비스에 대한 구성 (AS - IS)

Android App

Server

Activity

Service

onCreate()

서비스 연결

Dynamic BroadCastReceiver 생성

onResume()

Dynamic BroadCastReceiver 등록

onPause()

onStop()

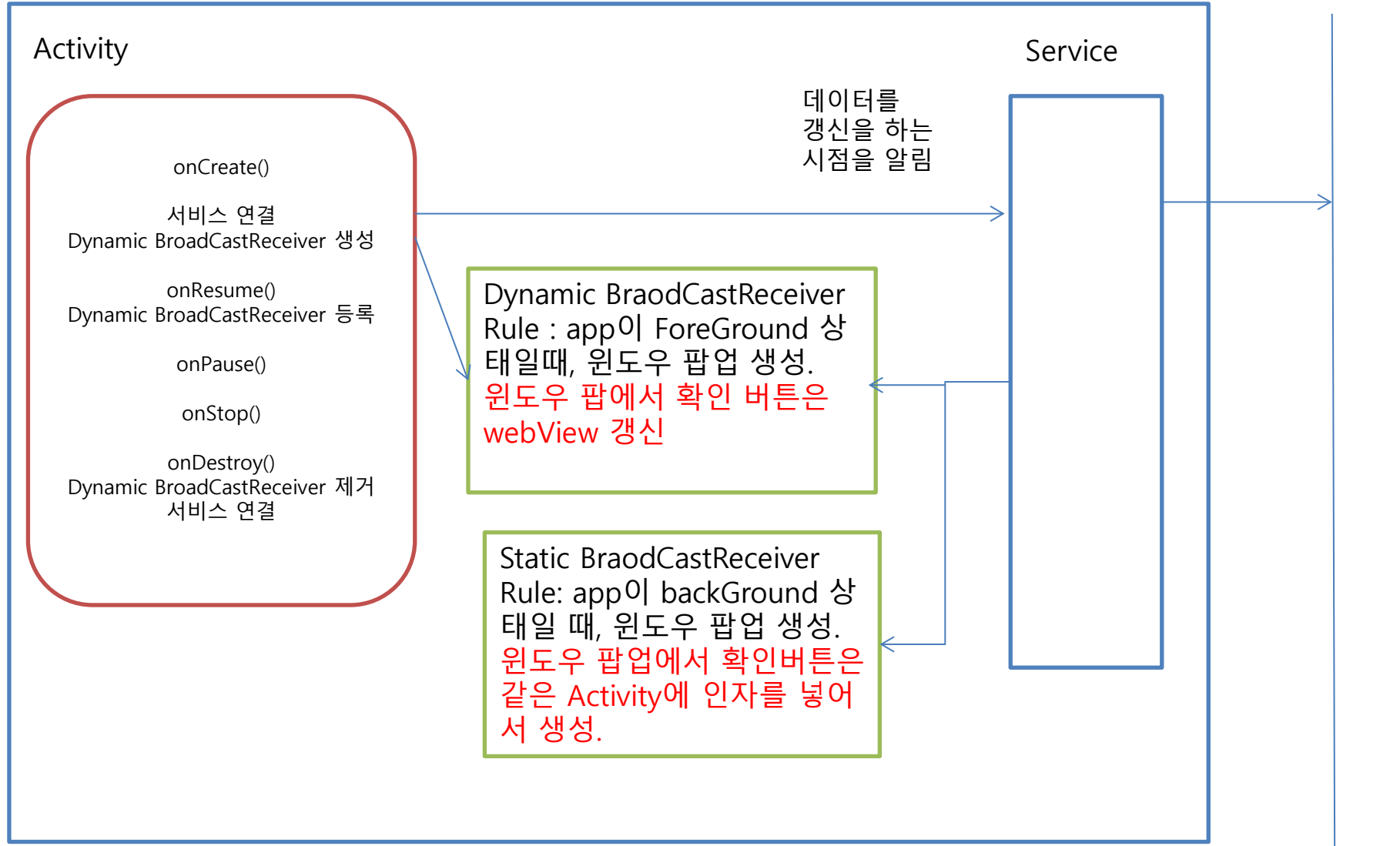
onDestroy()

Dynamic BroadCastReceiver 제거
서비스 연결

데이터를
갱신을 하는
시점을 알림

Dynamic BraodCastReceiver
Rule : app이 ForeGround 상
태일때, 윈도우 팝업 생성.
윈도우 팝업에서 확인 버튼은
webView 갱신

Static BraodCastReceiver
Rule: app이 backGround 상
태일 때, 윈도우 팝업 생성.
윈도우 팝업에서 확인버튼은
같은 Activity에 인자를 넣어
서 생성.



Android BroadCast의 이해 (referenced <http://abydos.tistory.com/26>)

- **Android에서 Application간 통신하거나 data를 공유하는 가장 간단한 방법은 Braodcast를 이용하는 것이다.**

1. 구성

- 구성은 BroadCast를 보내는 Sender와 수신하는 Receiver로 구성된다.
- Sender와 Receiver는 직접적인 연관성이 없기 때문에, Application에서 구현 될 수도 있지만, 서로다른 Application에서 구현 할 수도 있다.
- Broadcast 전달은 Binder를 이용한 IPC로 이루어진다. 그리고 내용은 Intent를 이용하게 된다.
- Intent를 이용하기 때문에 용량은 프로세스당 할당된 용량으로 제한된다. 할당된 용량은 1MB이다.
- 이런 제약조건때문에 이미지 같은 큰데이터는 Intent에 경로를 담아서 옮기게 된다.

2. 전송방법 (전부 비동기적으로 보낸다.)

- Context 의 sendBroadCast()와 sendOrderedBroadcast() api를 통하여 전송 할 수 있다.
- 1. sendBroadCast(Intent, permission)
 - 수신을 받기 위해서 Receiver에서 Action name에 해당하는 Intent filter를 추가해야된다.
 - Receiver가 구현된 Package에 Permission이 선언되어 있지 않다면, Braod cast가 보내지 않는다.
- 2. sendOrderedBroadcast()
 - 이전 것과 같이 비동기로 보내긴 하지만, 중요도에 따라 수신순서를 정하고 하나씩 전달된다.
 - 다시 말해 OnReceive()가 종료 되고 난 뒤에야 Reciever에게 BroadCast가 전달된다.

Android BroadCast의 이해 (referenced <http://abydos.tistory.com/26>)

3. 수신 방법

1. 정적인 선언에서 수신 방법

- BroadcastReceiver를 상속받는 클래스를 생성
- Manifest에 이전에 작성한 것을 Application 태그 안에
 - <receiver android:name= “패키지명.클래스명” ></receiver> 선언
- Intent-filter 선언 (receiver 태그 안에 다음과 같은 형태로 넣음 .
 - <receiver > <intent-filter> <action android:name= “액션Str” ></> </>
- 보통 액션은 “패키지명.ACTION_(원하는이름)” 형태로 정의가 된다.

2. 동적인 선언에서 수신 방법

- BroadcastReceiver를 상속받는 인스턴스 선언
- onReceiver 콜백 정의
- IntentFilter 인스턴스 생성 후 addAction 메소드로 받을 Intent들을 추가.
- onResume 시점에 이전에 생성한 BroadcastReceiver와, IntentFilter 인스턴스를 registerReceiver를 등록
- onDestroy 시점에 등록한 receiver를 unregisterReceiver로 해제 .

BroadcastReceiver (Dynamic)

- 현재 활성화된 UI(Activity)와 생애주기(LifeCycle)을 공유한다.
- App이 ForeGround일때, 주로 이용된다.
- 현재 활성화된 UI(Activity)를 조작할 수 있다.
- context 객체를 상속 받는 객체는 registerReceiver()를 이용하여 등록할 수 있다.
- context는 activity에게 상속한다.

BroadcastReceiver (Static)

- Application이 생성된 시점에 존재 한다.
- App이 Background에 있더라도, 이벤트 전달이 가능하다.
- 현재 활성화된 UI(Activity)를 알기 힘들다.
- Manifest 에 Tag형태로 선언한다.

Service 생성 옵션 referenced

<http://arabiannight.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9CAndroid-Service-%EC%82%AC%EC%9A%A9%EB%B2%95>

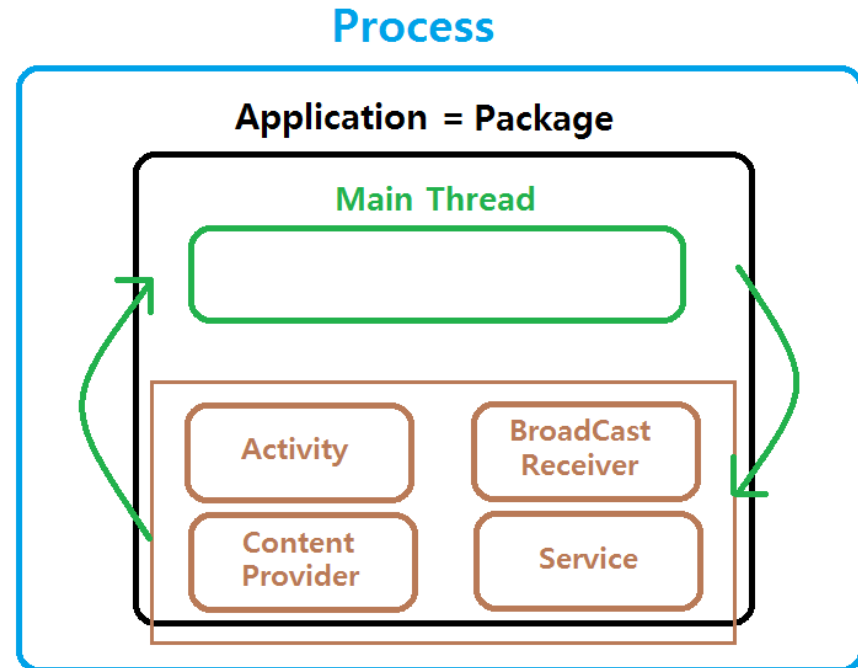
- **안드로이드 Application의 4대 컴포넌트 중하나임**
 - Activity, BroadcastReceiver, ContentProvider, Service
- **서비스란 ??**
 - Activity 처럼 사용자와 상호작용이 아닌, Background 에서 동작하는 컴포넌트이다.
- **서비스는 왜 필요한가.**
 - Activity가 종료되더라도, 동작하기 위해 만들어진 컴포넌트입니다. 예를들면 MP3
 - 만약 서비스가 실행된 상태라면, 안드로이드 OS에서는 Process를 웬만하면 죽이지 않습니다. 결과적으로 안드로이드 OS는 Process가 죽는 것을 방지하고 관리하게 됩니다.
- **서비스의 사용방법**
 - startService()
 - context.startService(Intent) 로 실행시키는 방법.
 - bindService()
 - Connection 객체를 이용하여 서로의 Data를 공유하기 위해 필요한 경우.

Service 생성 옵션 referenced

<http://arabiannight.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9CAndroid-Service-%EC%82%AC%EC%9A%A9%EB%B2%95>

• 서비스 사용시 주의사항

- 안드로이드는 기본적으로 리눅스의 커널에 기반함.
- 따라서, 커널들이 각 프로세스를 관리함.
- 각 프로세스는 MainThread하에 Service또한 관리됨. (보통은 UI작업을 처리해주는 스레드라 알려짐)
Service역시 MainThread에서 관리해야하므로 Thread작업이 필요할때 작업 Thread를 생성해서 관리해야한다.
- 다시말하면 어떤 Thread 작업이 필요하다면 별도의 Thread작업을 이용해야 한다 라는 것이고, 너무 오래걸리게되면 ANR에러를 유발할 것이다.



Service 생성 옵션 referenced

<http://arabiannight.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9CAndroid-Service-%EC%82%AC%EC%9A%A9%EB%B2%95>

- **Service 실행 중에도 startService 호출 (2.0 이상부터는 서비스의 문제점과 기능을 보완하였다.)**
 - Service 실행 중에도 startService를 호출하면 어떤 현상이 벌어질까? 그렇게 되면 onStartCommand를 호출하게 된다.
 - Service는 onCreate() -> onStartCommand() 주기부터 실행하게 된다. (Activity의 onResume() 과 유사함)
 - onStartCommand 시점부터 부터 호출 되기 때문에 서비스를 다시 초기화 해야만 한다거나 반복적으로 일어날 경우는 이 시점에 초기화를 해주도록 합시다.
 - 2.0 이상부터는 onStartCommand에서 초기화 하는 것을 권장한다.
- **onStartCommand의 메서드 리턴 타입에 대한 정의**
 - START_STICKY: service가 종료된 경우 시스템이 다시 Service를 재시작하지만 Intent값만 null로 초기화 시킨다.
 - START_NOT_STICKY : 강제로 종료된 Service가 재시작 하지 않스비다. 시스템에 의해 강제 종료되어도 괜찮은 작업을 진행할 경우 사용해 주시면됩니다.
 - START_REDELIVER_INTENT: START_STICKY와 마찬가지로 intent값을 그대로 유지시켜줍니다. startService() 메서드 호출시 Intent value 값을 이용한 경우 해당 flag를 사용하여 리턴값을 설정한다.

Android 주기적인 조회 서비스에 대한 구성 (TO - BE)

Android App

Server

Activity

Service

onCreate()
서비스 연결
Dynamic
BroadCastReceiver 생성
onResume()
Dynamic
BroadCastReceiver 등록
onPause()
onStop()
onDestroy()
Dynamic
BroadCastReceiver 제거
서비스 연결

Static BraodCastReceiver
Rule: app이 backGround 상
태일 때, 윈도우 팝업 생성.

갱신을 하는
시점을 알림

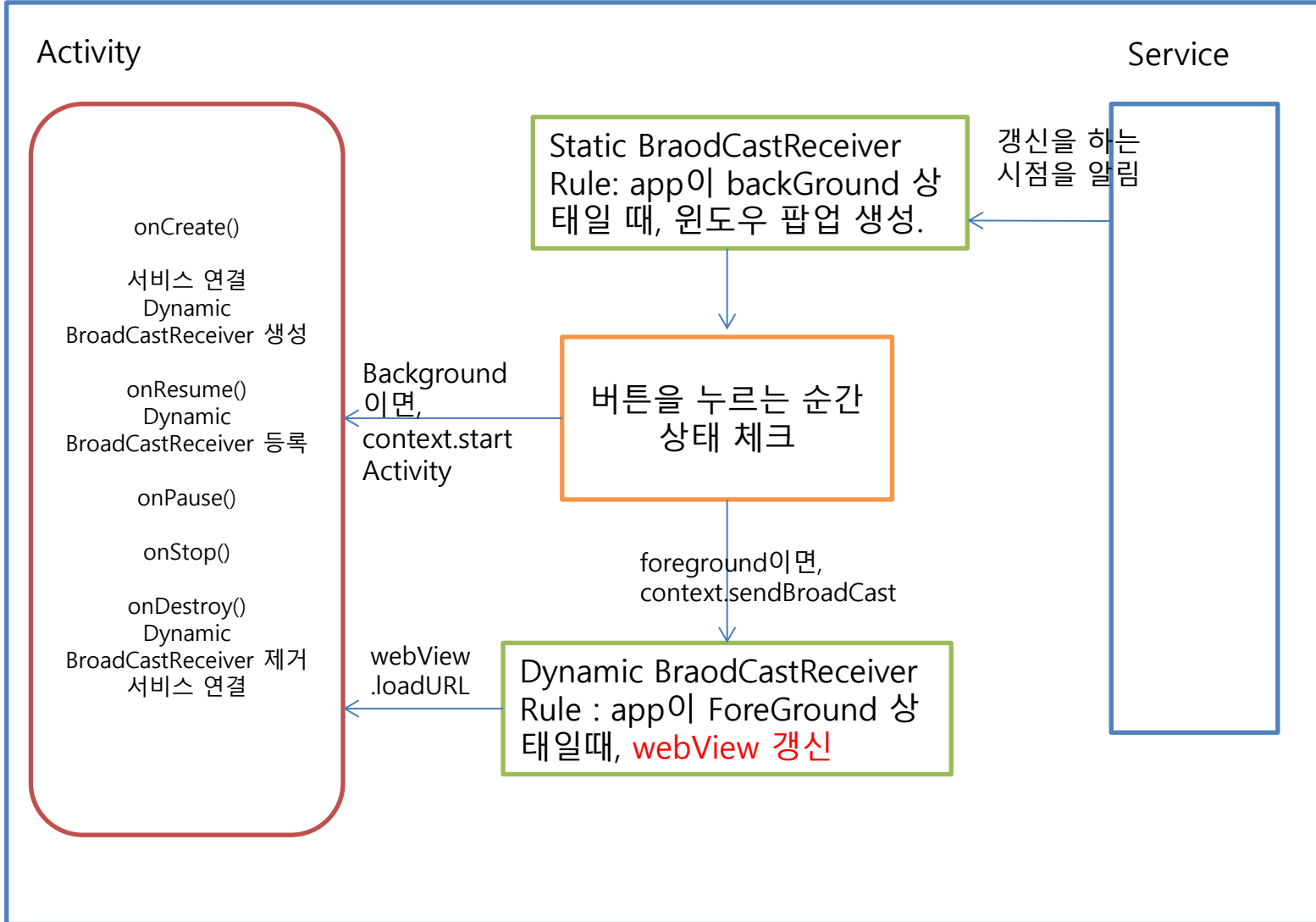
Background
이면,
context.start
Activity

버튼을 누르는 순간
상태 체크

foreground이면,
context.sendBroadCast

webView
.loadURL

Dynamic BraodCastReceiver
Rule : app이 ForeGround 상
태일때, **webView 갱신**



Context 란 ...?

- <http://blog.naver.com/PostView.nhn?blogId=huewu&logNo=110085457720&parentCategoryNo=&categoryNo=18&viewDate=&isShowPopularPosts=false&from=postView>