



# Started Service(실습)

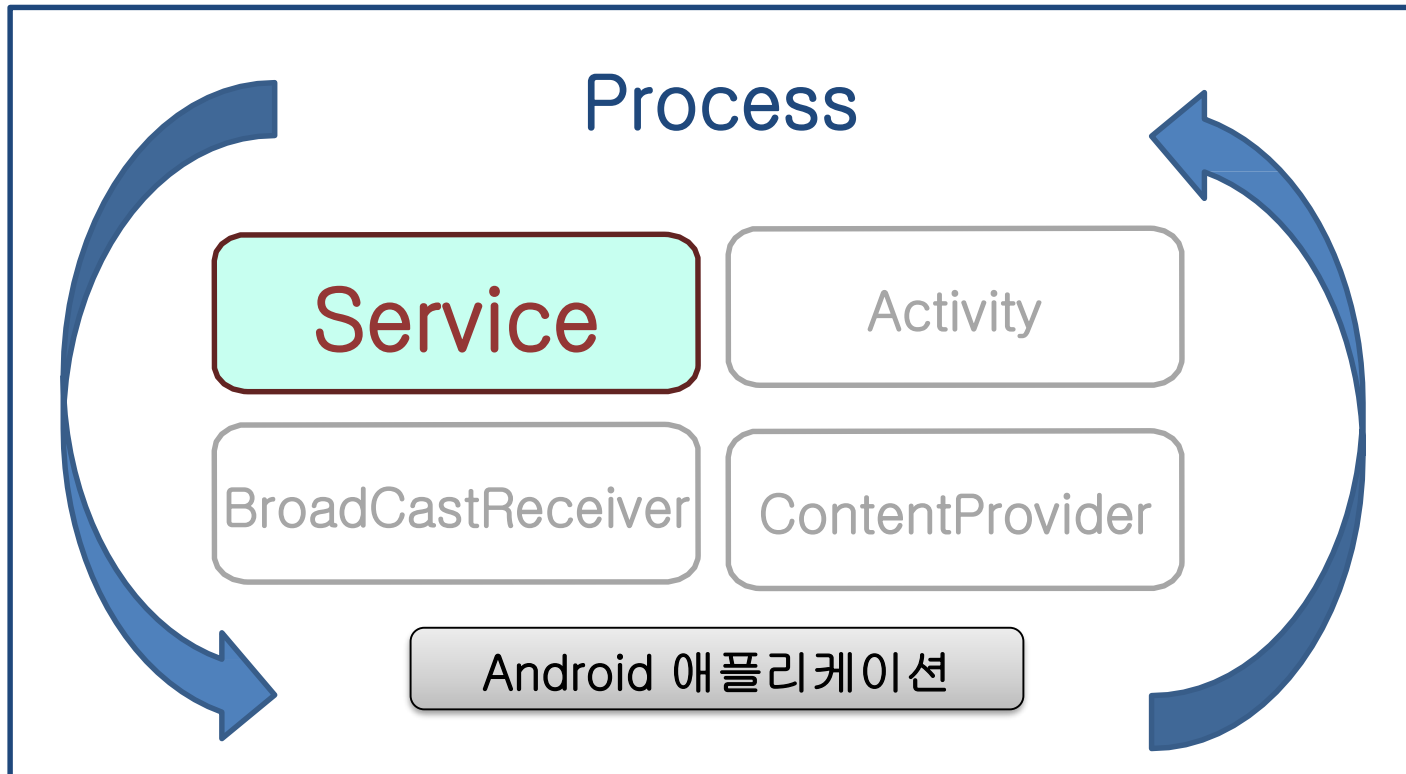
---

배 희호 교수  
경북대학교  
스마트IT과



# Service

- Service는 Android Application을 구성하는 4가지 Component중 하나이며, Activity처럼 사용자와 상호작용 하는 Component가 아니고, **Background(화면 뒷단)에서 동작 하는 Component**를 말함





# Service

## ■ 서비스 예





# Service



- Service = Deamon = Background Program
- Background에서 실행되는 정해지지 않은 시간동안 항상 동작하는 Process
- 사용자와 직접 상호작용은 하지 않음으로 **Visual한 사용자 인터페이스가 없음 (UI가 없음)**
- 사용자의 입력과는 무관하게 지속적인 처리나 규칙적인 처리, 또는 Event 처리를 수행하는 작업의 경우에 Service를 이용하면 좋음
  - 예) MP3 Player, 채팅 프로그램, 파일 다운로드 등...
- 사용자와 통신할 수 있는 방법이 필요
  - Notification
- Service는 일반적으로 애플리케이션에 의하여 시작됨
- Activity와 연결되어 있지 않아도 끊임없이 실행이 됨
- 연산이나 메소드 등의 서비스를 제공하는 것이 주 임무



# Service

- 사용자 인터페이스(UI) 없이 Background에서 실행되는 Component
  - 배경 음악을 재생
  - Web Site에서 주기적으로 데이터를 읽음
  - 주기적으로 폰의 사용량을 계산
  - 애플리케이션의 업데이트를 주기적으로 검사

서비스는 우리가 식사할 때 음악을 연주해주는 연주자들과 같은 존재이다.



액티비티





# Service

- Android를 개발하면서 아마도 가장 많이 사용되는 Component를 Activity로 알고 있겠지만, 사실 시스템 측면에서는 Service 임
- Android 시스템 자체가 대부분 Service로 이뤄져 있고, 심지어 Activity가 돌아가기 위해서도 Service를 사용하기 때문 임
- 하지만 우리는 이러한 Service가 존재하는 것을 전혀 인지하지 못한다. 왜일까?
- 바로 눈에 보이지 않고 Background에서 동작하기 때문 임
- Service는 Activity와 달리 화면을 가지지 않고 오직 Background에서만 동작
- Activity 애플리케이션은 화면(액티비티)이 종료되면 동작하지 않지만 Service는 Background에서 실행되므로 화면과 상관없이 계속 동작함



# Service



- Service 활용사례
  - 별도의 Thread를 통해서 사용 가능
    - 파일 다운로드
    - Media Player
- Android 소프트웨어 스택 내의 서비스 사용
  - Location Manger
  - Media Controller
  - Alarm Manager



# Service



- Android Service의 시작
  - 수동으로 시작하기
    - API를 호출
  - 자동으로 시작하기
    - IPC(프로세스간의 통신)를 사용
- Android Service의 주의사항
  - Service는 더 이상 필요 없을 때까지, 또는 메모리를 더 많이 확보해야 할 때까지 실행
  - 이로 인해 CPU와 Network(무선 랜)의 사용량이 많아져서 배터리의 사용시간이 줄어들 수 있음





# Service

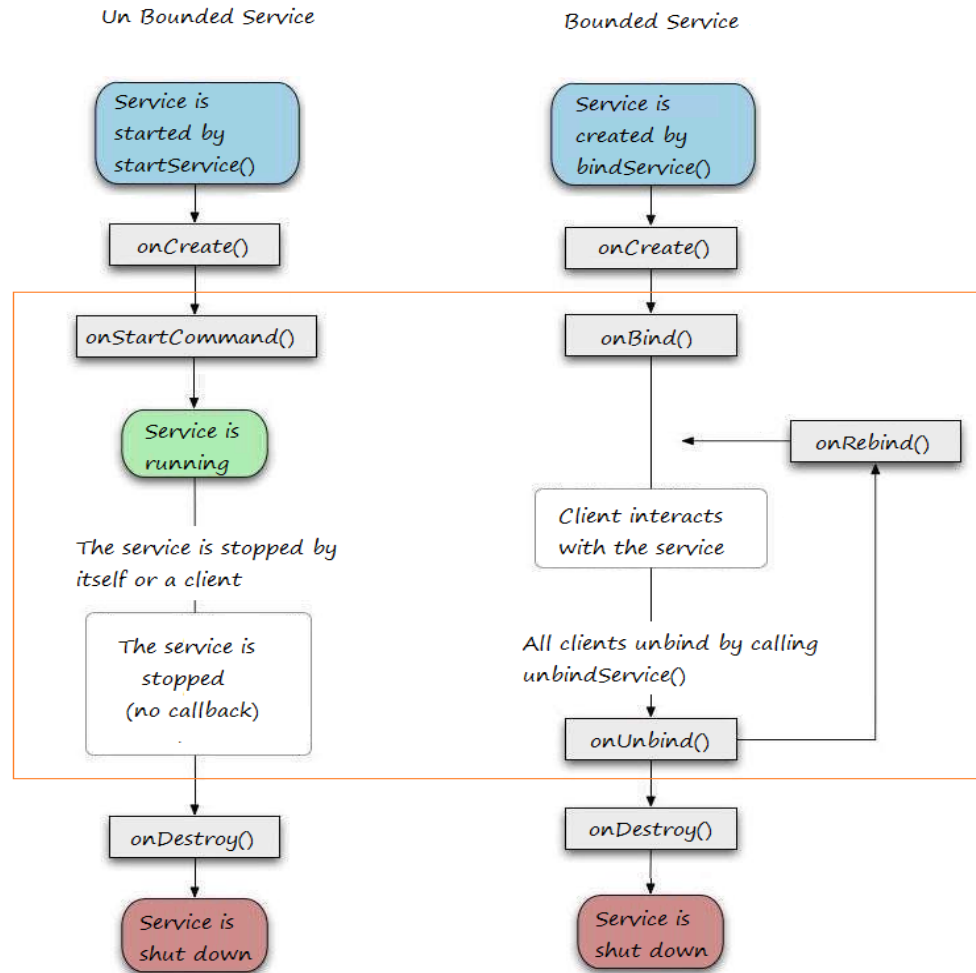


- Service 구현 과정
  - Android SDK에서 제공하는 서비스용 기본 클래스 상속
  - 새로운 서비스를 구현하는데 필요한 메소드를 오버라이드
  - AndroidManifest.xml 파일에 적절한 내용을 넣어 안드로이드 시스템과 연동
- Android Service 클래스의 구현은 Activity의 구현과 비슷함



# Service

## Service LifeCycle

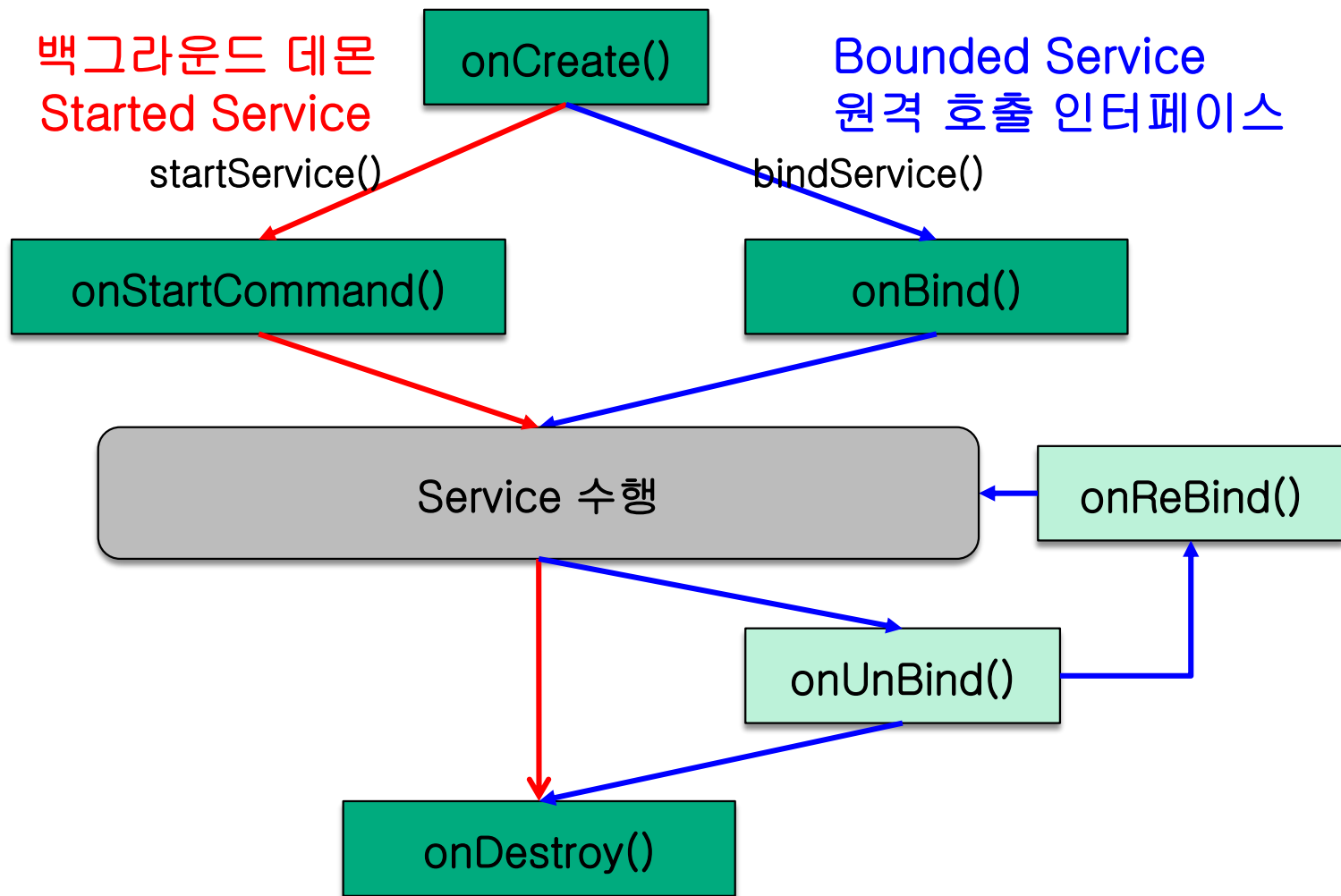




# Started Service

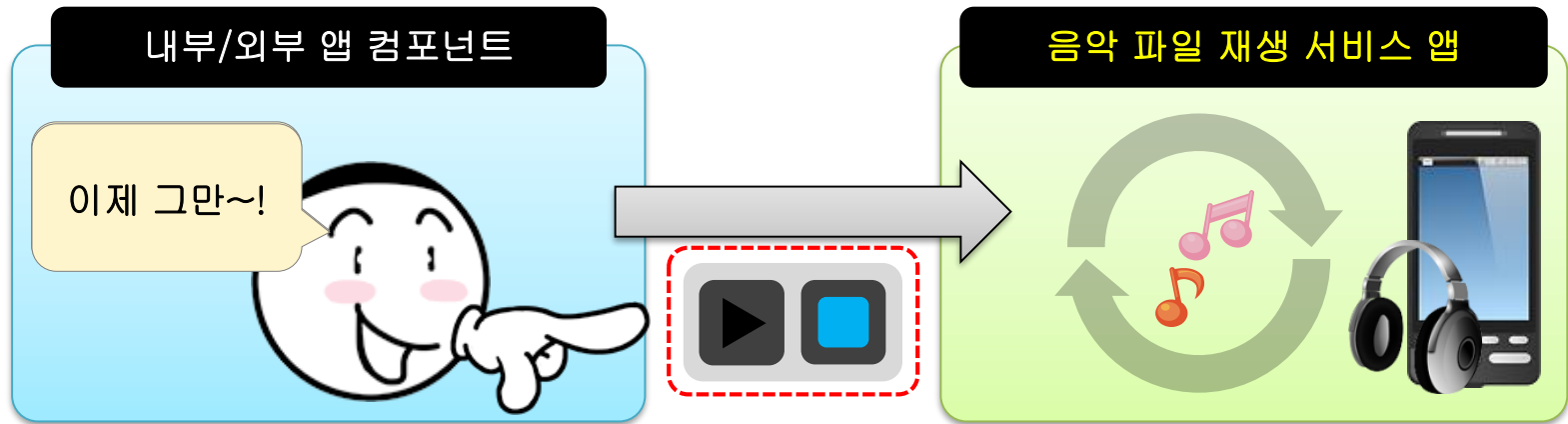
백그라운드 데몬  
Started Service

Bounded Service  
원격 호출 인터페이스





# Started Service



- 내/외부 앱과 음악 파일 재생 Service의 관계는 서로 독립적
- Service를 동작시킨 Component가 종료되더라도 Service는 작업이 끝날 때까지 동작을 유지



# Started Service



- 특정 Service를 Background로 동작시키는 것에 목적을 둔 형태
- Service를 발생시킨 곳에서는 중단시키는 것 이외에는 어떤 제어도 불가능
- 발생시키는 곳과 Service는 독립적이므로 Activity가 종료되어도 동작함
- Service가 실행되면 onCreate - onStartCommand 생명주기 함수가 호출되고, 종료될 땐 onDestroy 생명주기 함수가 호출
- 결국 Service가 동작 중일 때는 onStartCommand의 코드가 계속해서 동작하는 것



# Started Service



## ■ Service의 3가지 생명주기

```
void onCreate( )
```

```
void onStartCommand(Intent intent, int flags, int startId)
```

```
void onDestroy( )
```

### ■ onCreate()

- 첫 startService()에서만 호출되며, 이후에는 호출되지 않음

- 여기서 Intent를 다룰 수 없음

### ■ onStartCommand(Intent intent, int flags, int startId)

- Service를 외부에서 실행시키면 호출됨
- 다른 프로세스에서 Service를 수동으로 시작할 경우
- 여기서 Intent를 다루고 작업을 수행

### ■ onDestroy()

- 외부에서 stopService()를 호출하거나, 내부에서 stopSelf()를 호출 시 호출됨



# Started Service

- 이미 시작된 Service는 onStartCommand( ) 메소드의 실행을 마친 후 바로 onDestroy( )로 가는 것이 아님
- 다음 실행을 위해 대기 상태로 들어가기 때문에 프로세스는 계속 살아있게 됨
- 이때 외부에서 Service를 실행시키면 onStartCommand() 메소드가 호출됨
- 시스템에 리소스가 모자라게 되면, OS가 프로세스를 죽이고 이때 onDestroy() 메소드에 의해서 Service가 종료됨
- Service를 다시 시작하게 되면 onCreate( )가 작동하게 됨

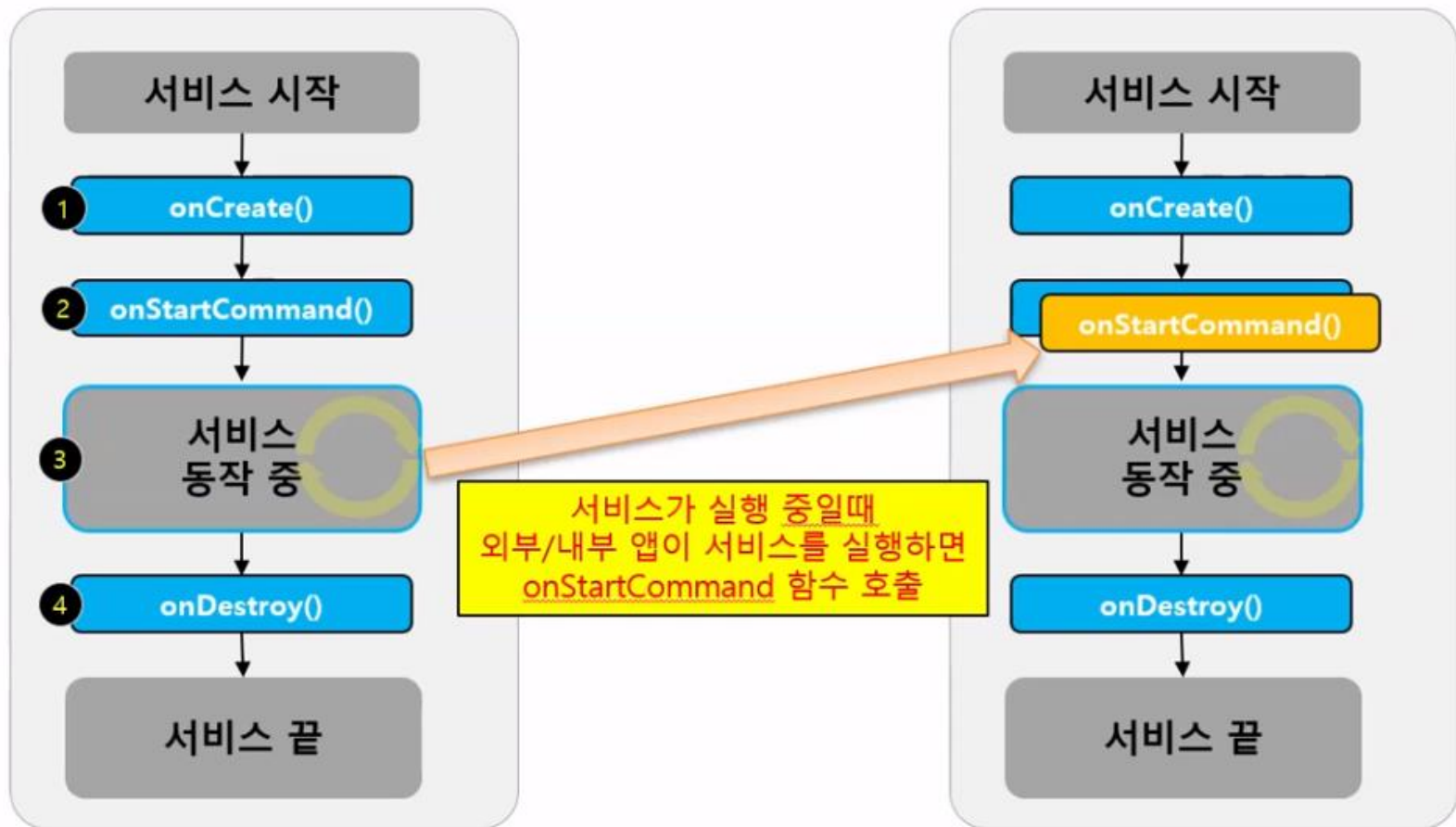
onCreate

onStartCommand

onDestroy



# Started Service







# Started Service



## ■ Start Service 생성하기

- Start Service는 `startService()` 메소드를 호출하여 시작할 수 있는 Component이며, Service의 `onStartCommand()` 메소드의 호출로 Service 이어짐
- Service가 시작되면, Service는 자신만의 생명주기를 갖고, Service를 실행시킨 Component가 종료되더라도 Background에서 계속 작동됨
- Service는 자신의 작업을 다 마무리 한 뒤, `stopSelf()` 메소드를 호출하여 종료될 수도 있고 또는 다른 Component에서 `stopService()`를 호출해서 종료 시킬 수도 있음
- Activity와 같은 Component는 `startService()` 메소드를 호출해서 Service를 시작시킬 수 있으며, 사용하고자 하는 Service 정보와 Service에서 사용될 데이터를 Intent에 담아 전달할 수 있음



# Started Service

- Service 만들기

- Service를 정의하려면 Service를 확장하는 새로운 클래스를 만들고 4개의 메소드를 재정의

- void onCreate()

- Service가 처음 생성될 때 호출되는 메소드

- 이 메소드 안에서 초기의 설정 작업을 하면 되고, Service가 이미 실행 중이면 이 메소드는 호출되지 않음



# Started Service

## ■ Service 만들기

- `int onStartCommand(Intent intent, int flags, int startId)`
  - 다른 Component가 `startService()`를 호출해서 Service가 시작되면 이 메소드가 호출
  - 만약 Bind Type의 Service(Bound Service)를 구현한다면, 이 메소드는 재정의 할 필요는 없음
  - Service가 실행되고 있는 상태에서 또 Service 시작을 할 경우 `onStartCommand()` 호출
  - 여기서 실제로 처리를 수행할 Background Thread를 띄우는 작업을 함
- 매개 변수
  - `intent` : Client가 Service를 시작할 때 전달하는 것
  - `flags` : Service의 요청에 대한 추가 정보
  - `startId` : Service 요청에 대한 고유한 식별자



# Started Service



- `int onStartCommand(Intent intent, int flags, int startId)`
  - 리턴값을 통해 Service의 재 시작 방식을 제어할 수 있음

정의된 상수	설명
START_STICKY	<ul style="list-style-type: none"><li>✓ 표준 방식</li><li>✓ 서비스가 런타임에 의해 종료되면 항상 재 시작되며, 재 시작 될 때마다 <code>onStartCommand</code>가 호출된다. 이때 전달되는 <code>intent</code>는 null 임</li><li>✓ 지속적인 백그라운드 작업이 필요한 경우나 음악 재생 서비스 등에 적합한 방식</li></ul>
START_NOT_STICKY	<ul style="list-style-type: none"><li>✓ 서비스가 런타임에 의해 종료되어도 <code>startService</code>를 다시 호출하지 않으면 해당 서비스는 중지된다.</li><li>✓ 업데이트나 네트워크 폴링과 같이 규칙적인 처리를 다루는 서비스에 적합 (중지되어도 다음 예약 시점에 다시 호출됨)</li></ul>
START_REDELIVER_INTENT	<ul style="list-style-type: none"><li>✓ 서비스가 런타임에 의해 종료되면 <code>startService</code>를 다시 호출하거나, 프로세스가 <code>stopSelf</code>를 호출하기 전에 종료된 경우에만 재 시작 된다. 후자의 경우에는 <code>onStartCommand</code>가 호출되며, 처리가 덜된 초기 <code>Intent</code>가 전달됨</li><li>✓ 서비스가 요청 받은 명령을 반드시 처리완료 해야 하는 경우에 적합한 방식</li></ul>



# Started Service

- Int onStartCommand(Intent intent, int flags, int startId)
  - flags 인자를 통해 서비스가 어떻게 시작됐는지 알아낼 수 있음

flag	설명
START_FLAG_RETRY	재 시작 방식이 START_STICKY이면서 서비스가 비정상적인 종료를 당한 후에 재 시작되었음을 나타냄
START_FLAG_REDELIVER	재 시작 방식이 START_REDELIVER_INTENT이면서 서비스가 stopSelf를 호출하기 전에 비정상적인 종료를 당한 후에 재 시작되었음을 나타냄



# Started Service

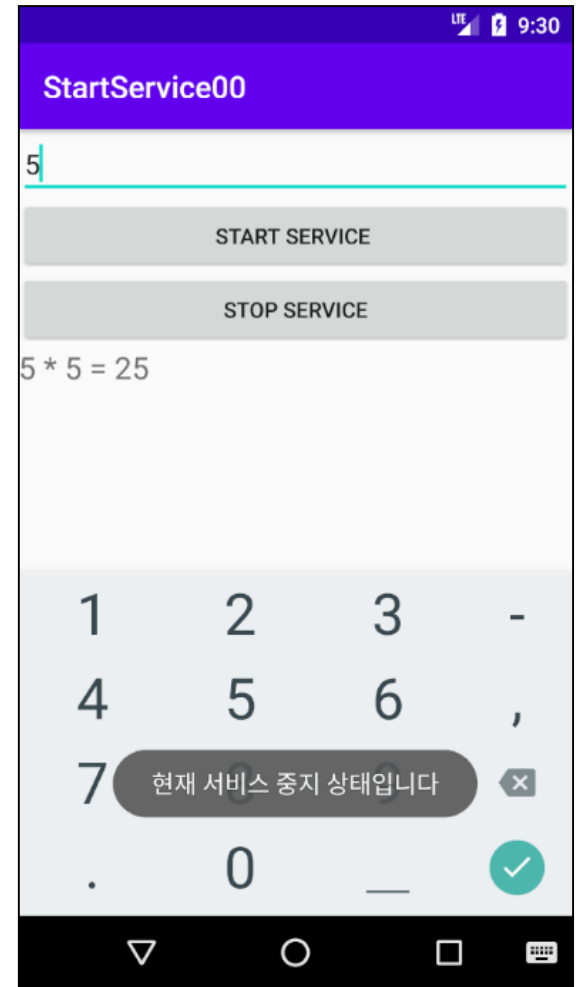
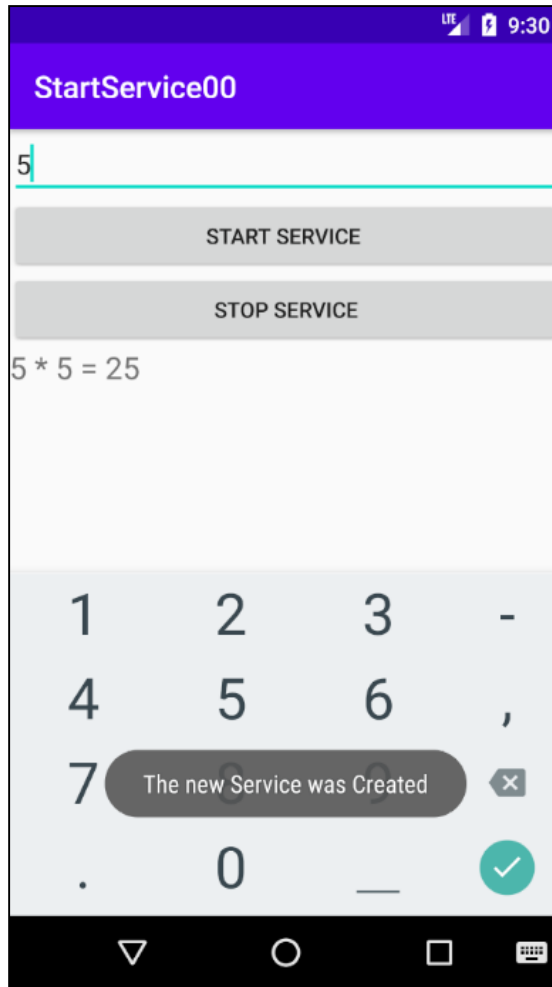
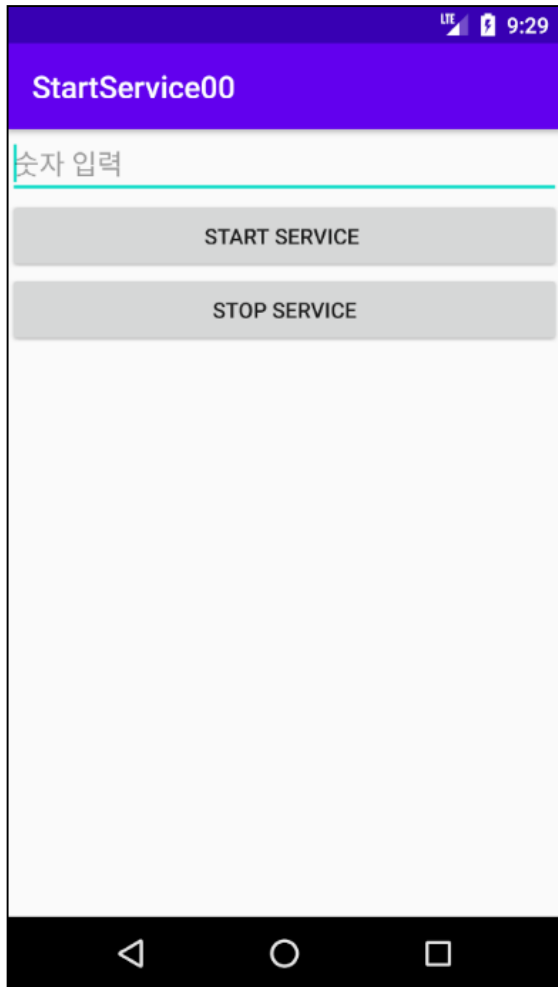
## ■ Service

- Android에서 사용되는 모든 Service의 기본이 되는 클래스
- 이 클래스를 상속받았을 때, Service의 모든 작업을 처리할 새로운 Thread를 생성하는 것이 중요함
- Service는 기본적으로 응용 프로그램의 Main Thread를 사용하기 때문에, 이를 그대로 사용하면 App에서 사용하는 Activity의 속도가 느려질 가능성이 큼
- onStartCommand() 메소드는 정수 값을 반환해야 함
- 이 정수 값을 어떻게 어떻게 설정하느냐에 따라서 시스템이 해당 서비스를 강제 종료했을 때, Service를 어떤 방법을 이용해서 다시 시작 시킬 지 결정할 수 있음



# Start Service 예제 0

■ 다음과 같은 Start Service를 만들어보자





# Start Service 예제 0

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="숫자 입력"
        android:inputType="number" />
```





# Start Service 예제 0

## ■ 사용자 인터페이스

<Button

```
    android:id="@+id/start"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Start service " />
```

<Button

```
    android:id="@+id/stop"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Stop Service" />
```

<TextView

```
    android:id="@+id/text"  
    android:textSize="20dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

</LinearLayout>



# Start Service 예제 0

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity
    implements View.OnClickListener {

    Intent intent;
    static TextView textView;
    EditText editText;
    static boolean isService = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = findViewById(R.id.edit);
        textView = findViewById(R.id.text);
        intent = new Intent(this, MyService.class);
        Button start = findViewById(R.id.start);
        Button stop = findViewById(R.id.stop);
        stop.setOnClickListener(this);
        start.setOnClickListener(this);
    }
```



# Start Service 예제 0

## ■ MainActivity.JAVA

@Override

```
public void onClick(View view) {  
    switch (view.getId()) {  
        case R.id.start:  
            if (isService) {  
                Toast.makeText(getApplicationContext(), "현재 서비스 중입니다.",  
                    Toast.LENGTH_SHORT).show();  
            } else {  
                String test = editText.getText().toString();  
                if (test.equals("")) {  
                    Toast.makeText(getApplicationContext(), "입력해주세요",  
                        Toast.LENGTH_SHORT).show();  
                } else {  
                    intent.putExtra("test", test);  
                    startService(intent);  
                }  
            }  
        }  
    }  
    break;
```



# Start Service 예제 0

## ■ MainActivity.JAVA

```
case R.id.stop:
    if (isService) {
        stopService(intent);
        Toast.makeText(getBaseContext(), "서비스 중지 합니다",
                        Toast.LENGTH_SHORT).show();
    }else
        Toast.makeText(getBaseContext(), "현재 서비스 중지 상태입니다",
                        Toast.LENGTH_SHORT).show();
    break;
}
}
```



The screenshot shows the Android Studio IDE. The top toolbar includes icons for running, debugging, and other development tools. The Project Explorer on the left shows the project structure with the 'com.example.service04' package selected. The 'New' context menu is open, displaying various options for creating new files and classes. The 'Service' option is highlighted in the 'New' submenu. The code editor in the background shows a snippet of Java code for a Service class, including the 'onBind' method and the 'create' method.



# Start Service 예제 0

## ■ Service 클래스의 상속을 받는 MyService 클래스 정의

New Android Component

Configure Component  
Android Studio

Creates a new service component and adds it to your Android manifest.

Class Name:

☒ Exported

☒ Enabled

Source Language:

Class Name must be unique

Previous Next Cancel Finish

- ✓ Exported는 현재 앱 외에 다른 앱이 이 서비스 구성 요소를 호출할 수 있다는 것을 의미
- ✓ Enabled는 이 서비스 구성 요소를 사용할 준비가 됨을 의미



# Start Service 예제 0

## ■ MyService.JAVA

```
public class MyService extends Service {
```

```
    @Nullable
```

```
    @Override
```

```
    public IBinder onBind(Intent intent) {
```

```
        return null;
```

```
    }
```

```
    @Override
```

```
    public void onCreate() {
```

```
        Toast.makeText(this, "새로운 서비스가 생성되었습니다",
```

```
                        Toast.LENGTH_LONG).show();
```

```
    }
```



# Start Service 예제 0

## ■ MyService.JAVA

@Override

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    MainActivity.isService = true;  
    Toast.makeText(this, " Service Started", Toast.LENGTH_LONG).show();  
    int value = Integer.parseInt(intent.getStringExtra("test"));  
    MainActivity.textView.setText(  
        String.format("%d * %d = %d", value, value, value * value));  
    stopSelf();  
    return super.onStartCommand(intent, flags, startId);  
}
```

@Override

```
public void onDestroy() {  
    super.onDestroy();  
    Toast.makeText(MyService.this, "Service Stopped",  
        Toast.LENGTH_SHORT).show();  
    MainActivity.isService = false;  
}  
}
```





# Start Service 예제 0

## ■ 메니페스트

```
android:theme="@style/AppTheme">
<service
    android:name="com.example.startservice00.MyService"
    android:enabled="true"
    android:exported="true"></service>

<activity android:name="com.example.startservice00.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>
```



# Start Service 예제 0

- startService()를 호출함과 동시에 Intent를 전달함으로써 Activity나 다른 Component에서 Service를 시작 시킬 수 있음
- Android System에서 Service와 관련된 onStartCommand()를 호출하고 그 메소드에 인텐트를 전달
  - 사용자가 직접 onStartCommand()를 호출할 수는 없음
- startService() 메소드는 호출되자마자 반환되며, Android System에서는 Service의 onStartCommand() 메소드를 호출
- Service가 생성된 상태가 아니었다면, 시스템은 onCreate() 메소드를 호출한 후에 onStartCommand() 메소드를 호출
- Service가 결속(Binding) 기능을 제공하지 않는다면, Service로부터 특정 결과값을 전달받는 것은 불가능



# Start Service 예제 0

- 결속(Binding) 기능 없이 Service로부터 결과값을 전달받길 원한다면 BroadCast와 관련된 PendingIntent를 생성한 뒤, Service를 시작시키는 startService()의 인자 값으로 전달하면 됨
  - Service가 자신의 작업을 다 끝낸 다음에, 결과 데이터를 전달하기 위해서 BroadCast 기능을 이용하면 됨
- Service가 시작되도록 요청은 여러 개가 존재할 수 있고, 그 결과 요청에 대응되는 onStartCommand() 메소드가 호출
- Start Service는 System에서 사용할 메모리가 부족하지 않는 한, System은 현재 작동하고 있는 Service를 절대 멈추게 하거나 종료시키지 않음
- Service를 종료하고자 할 때는 stopSelf()나 다른 Component에서 stopService() 메소드만을 한 번 호출함으로써 직접 Service를 종료해야만 함



# Start Service 예제 0

- stopSelf()나 stopService()가 호출되었을 때, System은 Service를 종료 시킴
- 만약 Service가 onStartCommand() 메소드에서 여러 개의 요청을 동시다발적으로 처리하고 있을 때, Service에 맨 처음 들어온 요청이 끝났다고 전체 Service를 정지시키면 안됨
  - 첫 번째 요청을 받은 이후에 두 번째 요청을 받았을 수도 있기 때문임
  - 처음 들어온 요청이 끝났다고 stopSelf()나 stopService()를 호출한다면 두 번째 요청이 처리되기도 전에 Service가 종료될 것임
  - 이런 문제를 피하기 위해서, stopSelf(int)라는 메소드를 호출함으로써 자신의 작업을 끝낸 Service 요청만 종료시킬 수 있음



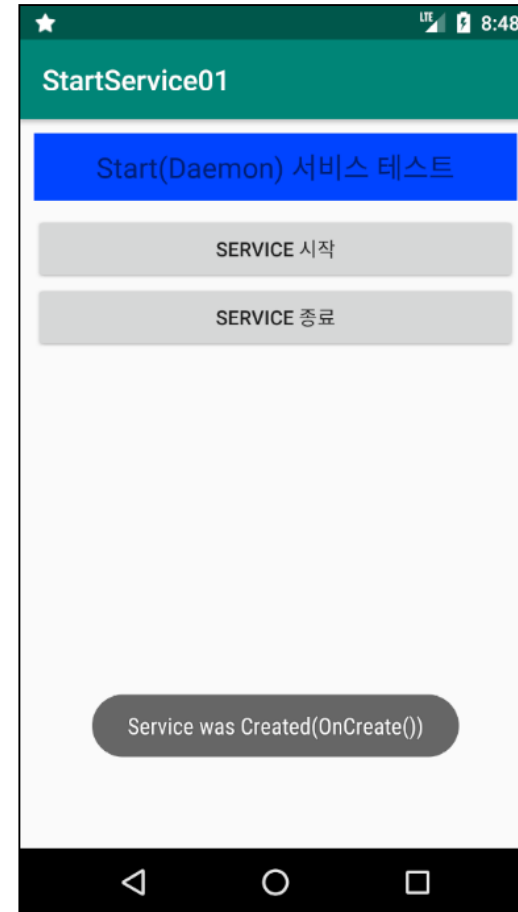
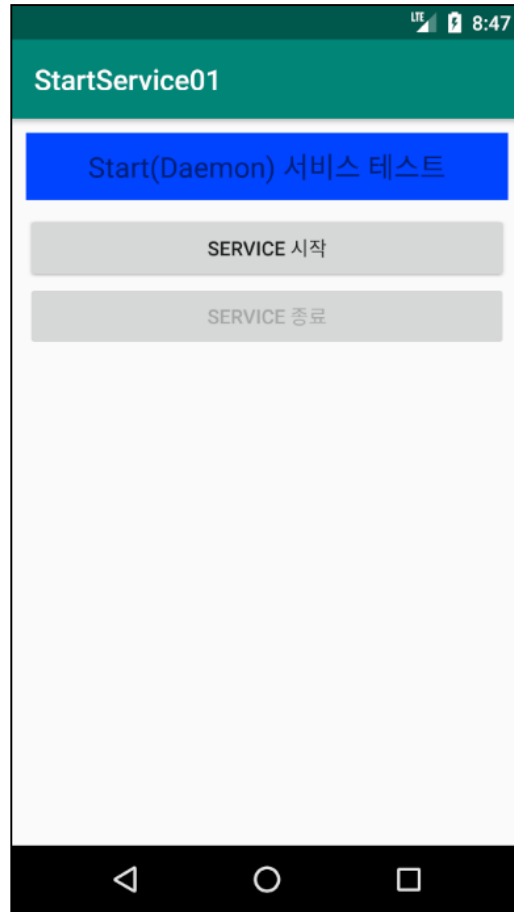
# Start Service 예제 0

- stopSelf(int)라는 메소드를 호출할 때, 해당 서비스 요청과 관련된 ID값을 인자로 전달하면 됨
  - ID값은 onStartCommand()에서 전달 받을 수 있음
- 두 번째 요청이 들어온 이후 작업을 끝낸 첫 번째 요청을 stopSelf(int) 메소드로 종료해도, 두 번째 요청은 첫 번째 요청과 ID값이 다르기 때문에 종료되지 않음



# Start Service 예제 1

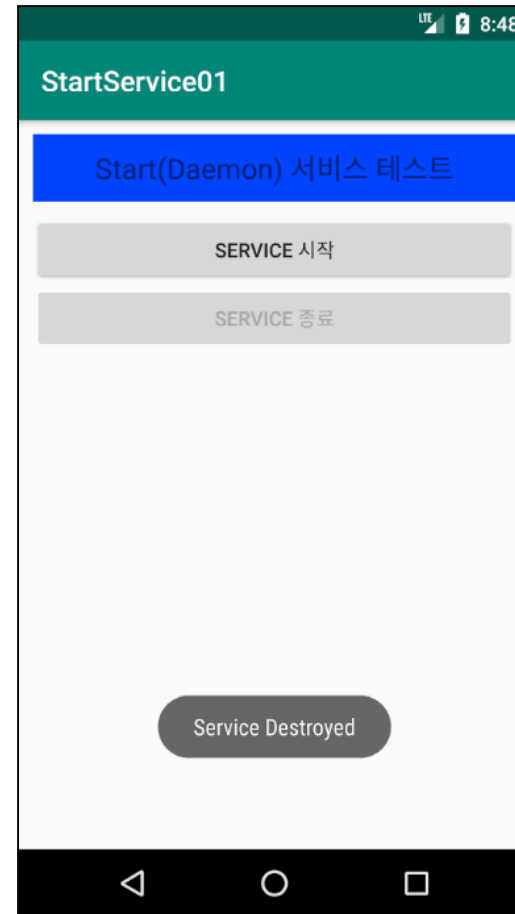
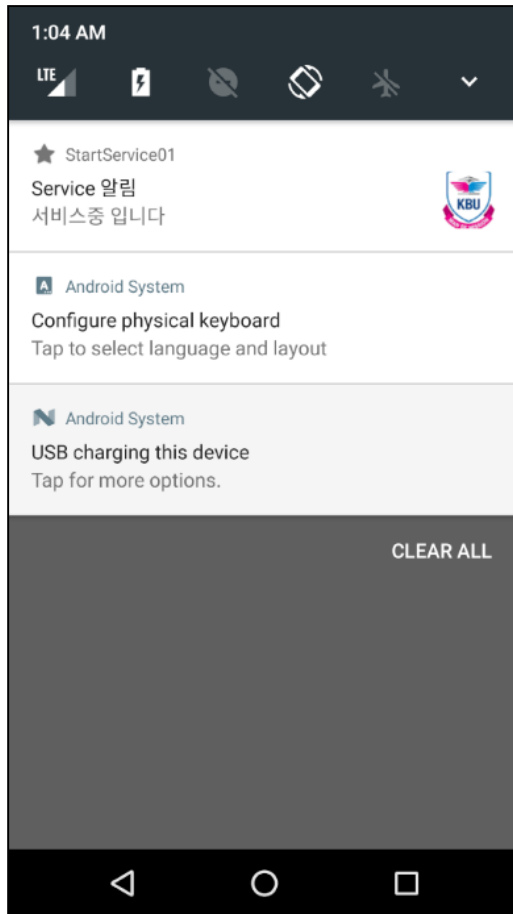
- 다음과 같은 Start Service를 구현하여보자





# Start Service 예제 1

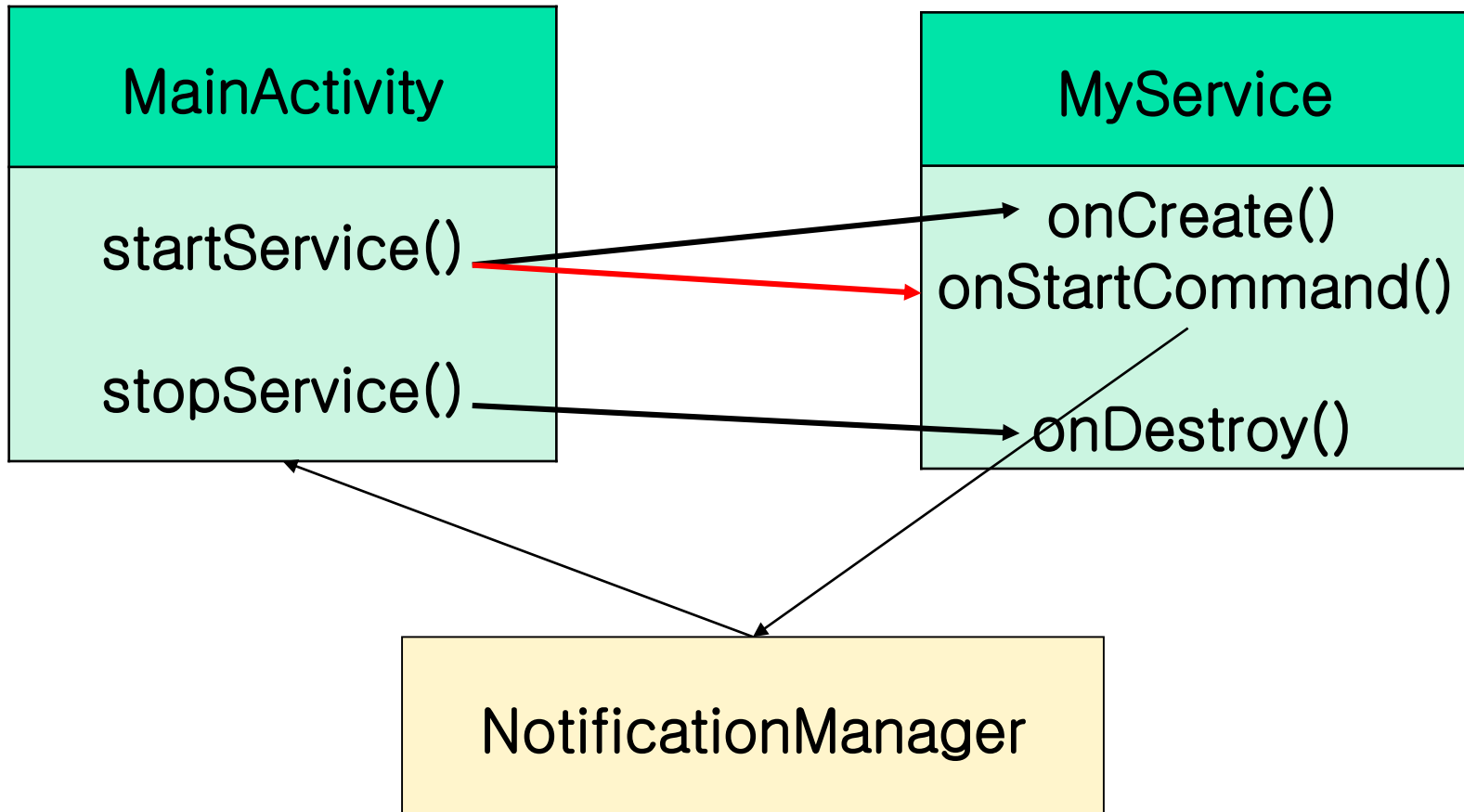
## ■ 실행 결과





# Start Service 예제 1

## ■ 서비스 생성 및 제어







# Start Service 예제 1

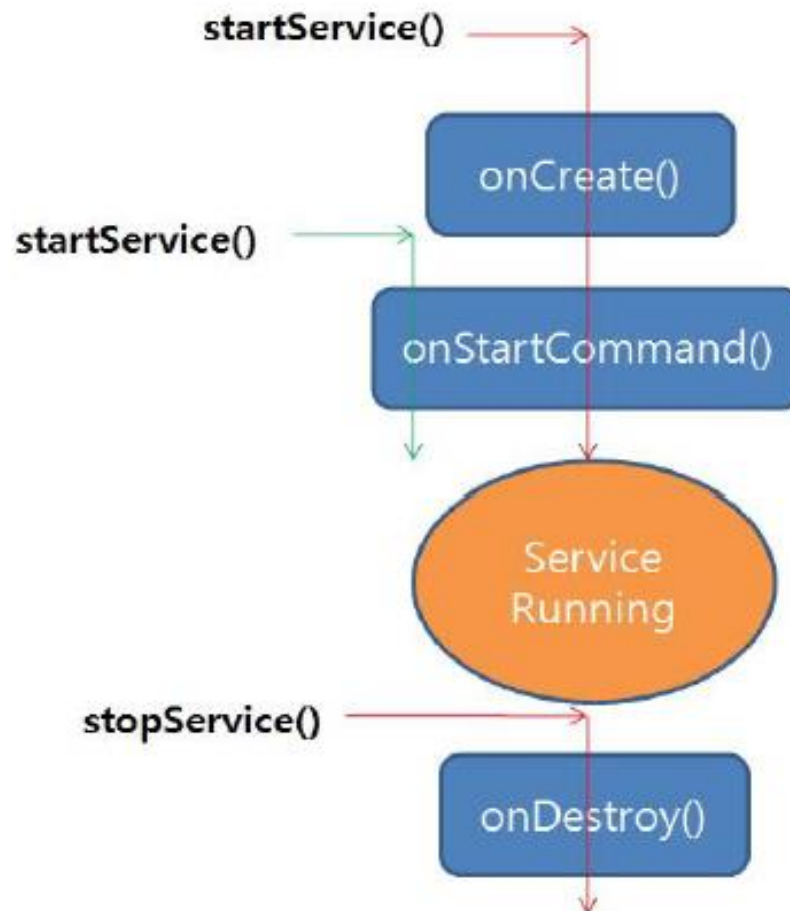


- Start Service 구현 순서
  - Service 시작, 종료 Button 레이아웃 구성하기
  - Service를 상속 받은 MyService class를 만듦
  - MyService의 onCreate(), onStartCommand() 메소드 재정의
  - Manifest에 새로 만든 Service 등록
  - MainActivity에서 startService()를 통해 MyService 시작, 한번 생성된 서비스를 다시 실행할 때는 onStartCommand() 실행
  - 여러 번 시작한 Service라도 stopService()로 한번 만에 종료



# Start Service 예제 1

## ■ startService( ) 메소드 이용 시 LifeCycle





# Start Service 예제 1

## ■ Service 시작하기

### ■ ComponentName startService(Intent intent)

```
Intent intent = new Intent(MainActivity.this, Myservice.class);  
startService(intent);
```

## ■ Service 중지하기

### ■ boolean stopService(Intent intent)

```
Intent intent = new Intent(MainActivity.this, Myservice.class);  
stopService(intent);
```

## ■ Service 스스로 종료하기

### ■ void stopSelf(int startId)

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    id = startId;  
    stopSelf(id);  
}
```



# Start Service 예제 1



## ■ StartService 실행 방법

암시적 서비스 실행	명시적 서비스 실행
<code>startService(new Intent(MyService.MY_ACTION);</code>	<code>startService(new Intent(this, MyService.class);</code>

- 암시적 Service를 위해서 MyService class에 MY\_ACTION을 포함 Intent 필더에 MY\_ACTION을 공급자로 등록 해야 함
- 안정성 높은 App을 구현하기 위해, Service를 시작하거나 Service와 결속할 때, 항상 명시적인 Intent를 이용해야 함
- 자신이 사용하고자 하는 Service의 이름을 명시적으로 정함으로써, 자신이 의도하지 않았던 Service가 실행되는 위험을 방지할 수 있기 때문임



# Start Service 예제 1

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="10dp"
        android:text="Start(Daemon) 서비스 테스트"
        android:background="#0044FF"
        android:textSize="20sp" />
```



# Start Service 예제 1

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/start"  
    android:layout_marginTop="10dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Service 시작"/>
```

```
<Button  
    android:id="@+id/stop"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Service 종료"  
    android:enabled="false"/>
```

```
</LinearLayout>
```



# Start Service 예제 1

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    Button stop;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```
        final Intent intent = new Intent(MainActivity.this, MyService.class);
```

```
        Button start = findViewById(R.id.start);  
        start.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View view) {
```

```
                startService(intent);
```

```
                stop.setEnabled(true);
```

```
            }
```

```
        });
```



# Start Service 예제 1

## ■ MainActivity.JAVA

```
stop = findViewById(R.id.stop);
stop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        stopService(intent);
        stop.setEnabled(false);
    }
});
}
```





# Start Service 예제 1

- Service를 시작하려면 Service를 지정할 Intent를 전달 인자로 하여 **startService()** 메소드를 호출

```
Intent intent = new Intent(this, MyService.class);  
startService(intent);
```

- Service를 정지시키려면 외부로부터 **stopService()** 메소드를 호출

```
stopService(intent);
```



# Start Service 예제 1

- Service 클래스를 상속하여 새로운 클래스를 정의
- 해당 메소드를 오버라이딩

```
public class MyService extends Service {  
    @Override  
    public void onCreate(){  
        //서비스가 생성 될 때 수행할 동작  
    }  
    @Override  
    public IBinder onBind(Intent intent){  
        //서비스 바인딩 구현으로 대체한다  
        return null;  
    }  
    @Override  
    public int onStartCommand(Intent intent, int flags, int startId){  
        return super.onStartCommand(intent, flags, startId);  
        //서비스가 시작될 때 수행할 동작  
    }  
    @Override  
    public void onDestroy(Intent intent, int startId){  
        //서비스가 종료될 때 수행할 동작  
    }  
}
```



# Start Service 예제 1

## ■ MyService.JAVA

```
public class MyService extends Service {  
    final int TEST_FLAG = 111;  
    NotificationManager manager;  
  
    @Nullable  
    @Override  
    public IBinder onBind(Intent intent) {  
        throw new UnsupportedOperationException("아직 구현되지 않았음");  
    }  
  
    @Override  
    public void onCreate() {  
        manager = (NotificationManager)  
            getSystemService(Context.NOTIFICATION_SERVICE);  
        Toast.makeText(getBaseContext(),  
            "Service was Created(OnCreate())", Toast.LENGTH_LONG).show();  
    }  
}
```



# Start Service 예제 1

## ■ MyService.JAVA

@Override

```
public void onDestroy() {  
    Toast.makeText(getBaseContext(), "Service Destroyed",  
                   Toast.LENGTH_LONG).show();  
    manager.cancelAll();  
}
```

@RequiresApi(api = Build.VERSION\_CODES.JELLY\_BEAM)

```
public int onStartCommand(Intent intent, int flags, int startId){  
    Toast.makeText(getBaseContext(),  
                   "Service was Created(OnStartCommand())", Toast.LENGTH_LONG).show();  
    manager.notify(TEST_FLAG, makeNotification());  
    return START_STICKY;  
}
```



# Start Service 예제 1

## ■ MyService.JAVA

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
private Notification makeNotification() {
    Notification.Builder builder = new Notification.Builder(this);
    Bitmap largelcon = BitmapFactory.decodeResource(getResources(),
        R.mipmap.kyungbok);
    builder.setContentTitle("Service 알림")
        .setContentText("서비스중 입니다")
        .setSmallIcon(R.drawable.btn_star_big_on)
        .setLargelcon(largelcon);
    return builder.build();
}
```



# Start Service 예제 1

## ■ 매니페스트에 <service> 등록

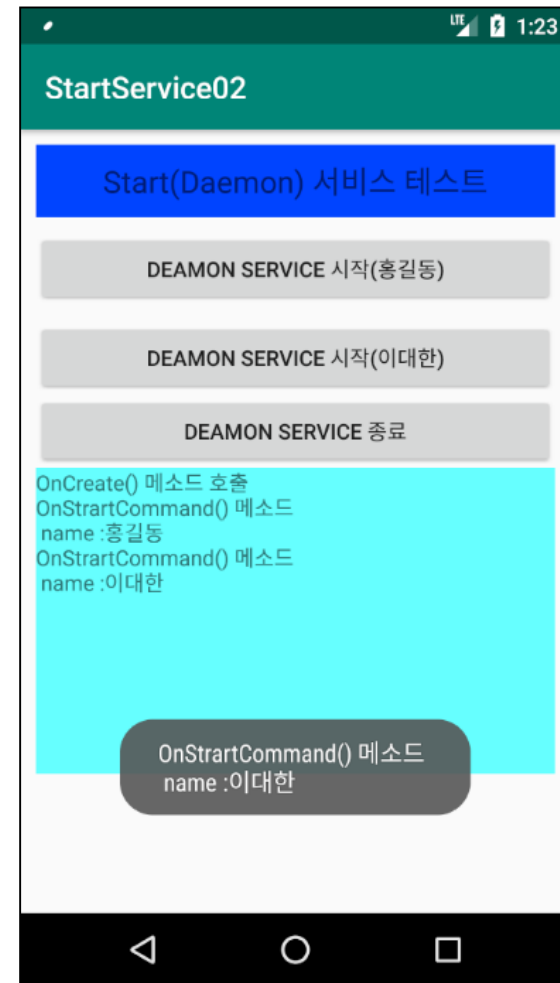
.....

```
<activity android:name="com.example.service01.MainActivity">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
  
    <category android:name="android.intent.category.LAUNCHER"/>  
  </intent-filter>  
</activity>  
<service  
  android:name="com.example.service01.MyService"  
  android:enabled="true"/>  
</application>  
  
</manifest>
```



# Start Service 예제 2

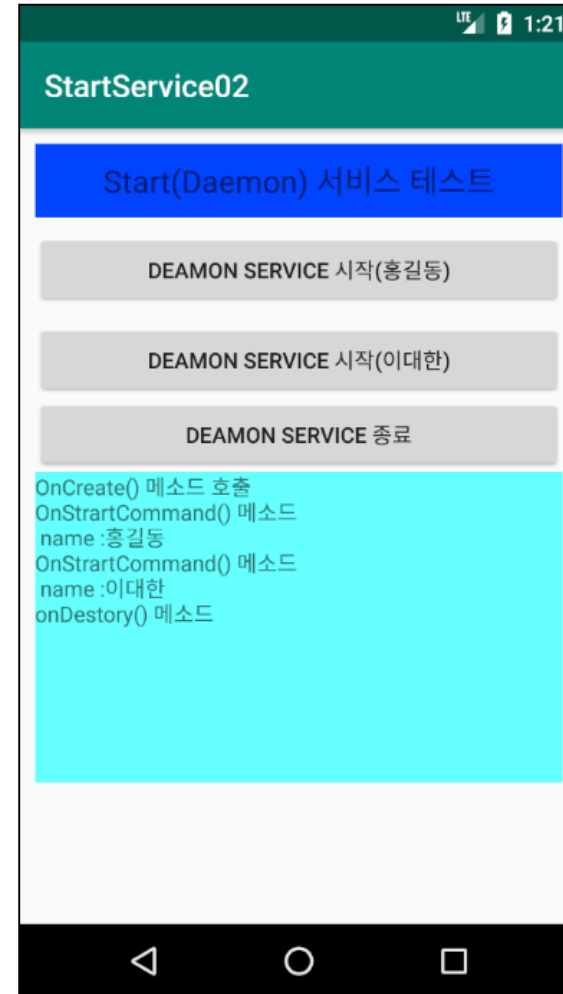
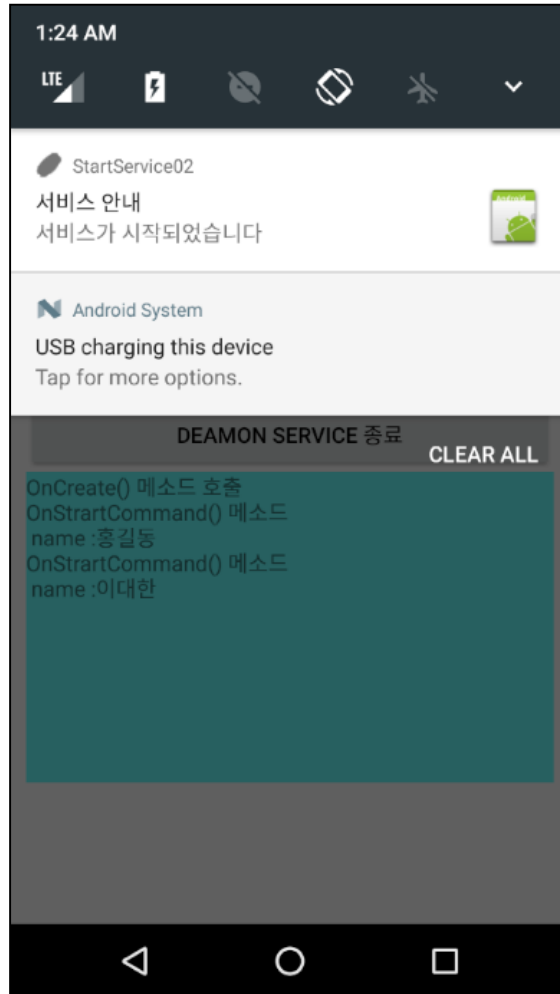
■ 다음과 같은 Start Service를 구현하여보자





# Start Service 예제 2

## ■ 실행 화면







# Start Service 예제 2

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="10dp"
        android:text="Start(Daemon) 서비스 테스트"
        android:background="#0044FF"
        android:textSize="20sp" />
```



# Start Service 예제 2

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/start1"  
    android:layout_marginTop="10dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Deamon Service 시작(이대한)"/>
```

```
<Button  
    android:id="@+id/stop"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Deamon Service 종료"/>
```



# Start Service 예제 2

## ■ 사용자 인터페이스

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:background="#66FFFF">

    <TextView
        android:id="@+id/result"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</ScrollView>
</LinearLayout>
```



# Start Service 예제 2

## ■ MyService 클래스

```
public class MyService extends Service {  
    private final int TEST_FLAG = 111;  
    NotificationManager manager;  
  
    @Override  
    public IBinder onBind(Intent intent) {  
        return null;  
    }  
  
    @Override  
    public void onCreate() {  
        manager = (NotificationManager)  
            getSystemService(Context.NOTIFICATION_SERVICE);  
        Toast.makeText(this, "OnCreate() 메소드",  
            Toast.LENGTH_SHORT).show();  
        MainActivity.textView.append("OnCreate() 메소드 호출\n");  
    }  
}
```



# Start Service 예제 2

## ■ MyService 클래스

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAM)
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    String name = intent.getStringExtra("name");
    Toast.makeText(this, "OnStartCommand() 메소드\n name :" + name,
        Toast.LENGTH_SHORT).show();
    MainActivity.textView.append("OnStartCommand() 메소드\n name :"+
        name + "\n");
    manager.notify(TEST_FLAG, makeNotification());
    return super.onStartCommand(intent, flags, startId);
}
```



# Start Service 예제 2



```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
private Notification makeNotification() {
    Notification.Builder builder = new Notification.Builder(this);
    Bitmap largelcon = BitmapFactory.decodeResource(getResources(),
                                                R.mipmap.icon);
    builder.setContentTitle("서비스 안내")
        .setContentInfo("Content Info")
        .setContentText("서비스가 시작되었습니다")
        .setProgress(100, 20, false)
        .setSmallIcon(R.mipmap.goldcoin)
        .setLargeIcon(largelcon);
    return builder.build();
}

@Override
public void onDestroy() {
    manager.cancelAll();
    Toast.makeText(this, "onDestroy() 메소드",
                    Toast.LENGTH_SHORT).show();
    MainActivity.textView.append("onDestroy() 메소드\n");
}
}
```



# Start Service 예제 2

- 다음과 같이 서비스 시작하기 버튼을 누르면 서비스 자신을 호출한 쪽에서 전달한 intent를 받을 수 있고 intent에 담겨진 값을 key, value형태로 해서 파라미터 값을 가져올 수 있음



# Start Service 예제 2

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    Button button1, button2, button3;  
    static TextView textView;
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    textView = findViewById(R.id.result);  
    final Intent intent = new Intent(getApplicationContext(), MyService.class);
```





# Start Service 예제 2

## ■ MainActivity.JAVA

```
button1 = findViewById(R.id.start);  
button1.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        intent.putExtra("name", "홍길동");  
        startService(intent);  
    }  
});
```

```
button2 = findViewById(R.id.start1);  
button2.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        intent.putExtra("name", "이대환");  
        startService(intent);  
    }  
});
```



# Start Service 예제 2

## ■ MainActivity.JAVA

```
button3 = findViewById(R.id.stop);  
button3.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        stopService(intent);  
    }  
});  
}
```



# Start Service 예제 2

## ■ 매니페스트에 <service> 등록

.....

```
<activity android:name="com.example.startservice02.MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER"/>
  </intent-filter>
</activity>
<service
  android:name=".MyService"
  android:enabled="true"/>
</application>

</manifest>
```



# Start Service 예제 3(음악 연주)

- 화면이 종료되어도 배경에서 음악을 연주하는 서비스를 작성하여 보자

서비스는 우리가 식사할 때 음악을 연주해주는 연주자들과 같은 존재이다.



액티비티

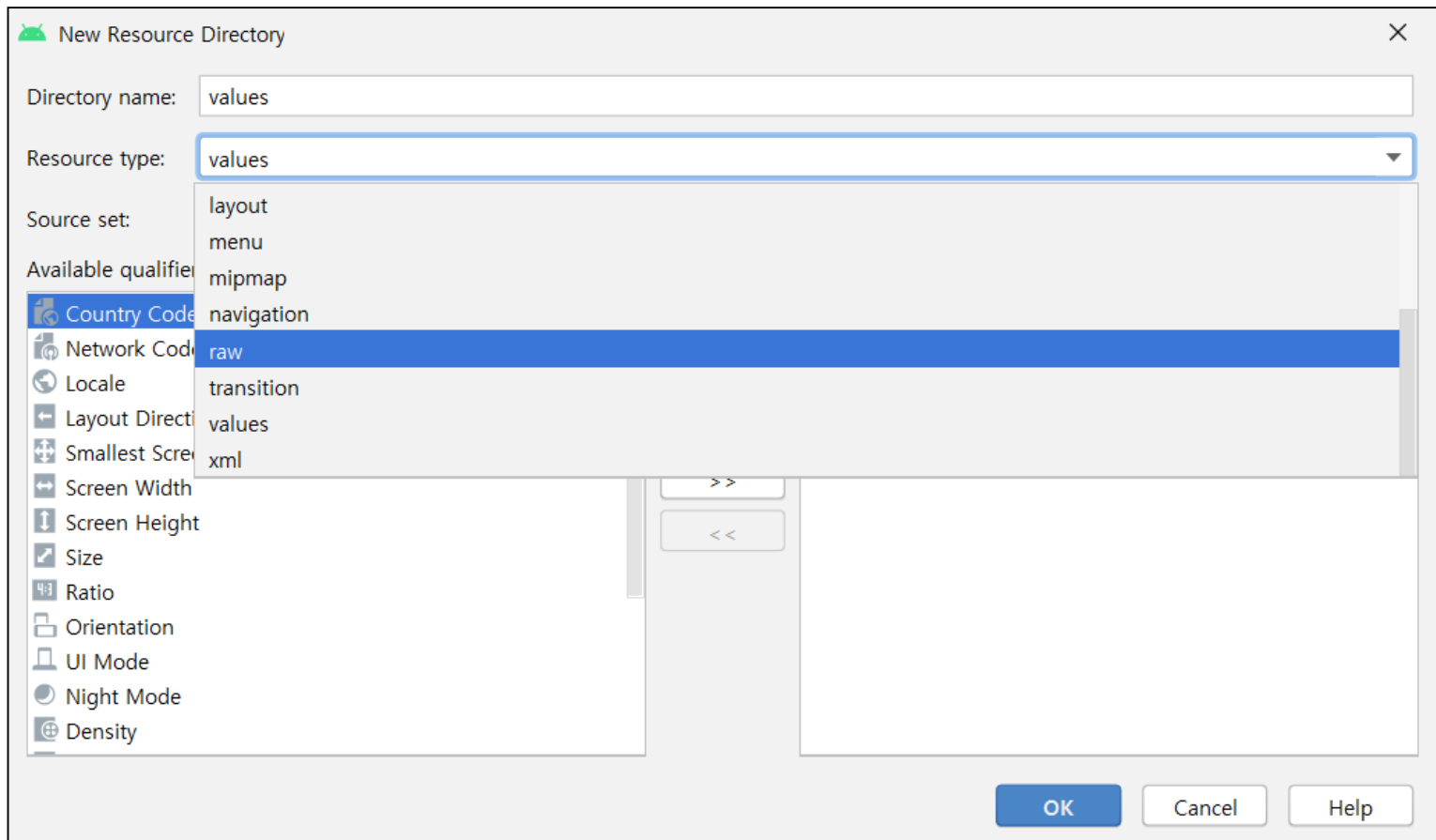




# Start Service 예제 3(음악 연주)

## ■ 음원 리소스 준비

- res 폴더 아래에 raw 폴더를 생성
- MP3 파일을 하나 복사





# Start Service 예제 3(음악 연주)

## ■ activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="10dp"
        android:text="음악 서비스 테스트(Start Service) "
        android:textColor="#FFFFFF"
        android:background="#0044FF"
        android:textSize="20sp" />
```



# Start Service 예제 3(음악 연주)

## ■ activity\_main.xml

```
<ProgressBar
    android:id="@+id/progress"
    android:padding="10dp"
    style="@style/Widget.AppCompat.ProgressBar.Horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<TextView
    android:id="@+id/status"
    android:gravity="center"
    android:text="00:00/00:00"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/start"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Music Service 시작"/>
</LinearLayout>
```



# Start Service 예제 3(음악 연주)

## ■ ProgressBar를 위한 Thread 작성

```
public class MyThread extends Thread {  
    MediaPlayer player;  
    int size;  
  
    public MyThread(MediaPlayer player, int size) {  
        this.player = player;  
        this.size = size;  
    }  
}
```





# Start Service 예제 3(음악 연주)

## ■ ProgressBar를 위한 Thread 작성

@Override

```
public void run() {  
    while (MainActivity.isPlaying) {  
        int current= player.getCurrentPosition();  
        final String msg = MusicService.getTime(current)+ "/"  
                                + MusicService.getTime(size);  
  
        Message message = MainActivity.handler.obtainMessage(0);  
        Bundle bundle = new Bundle();  
        bundle.putString("value", msg);  
        bundle.putInt("current", current);  
        message.setData(bundle);  
        MainActivity.handler.sendMessage(message);  
    }  
}
```



# Start Service 예제 3(음악 연주)

■ onCreate( ), onDestroy( ), onStartCommand( ) 메소드

```
public class MusicService extends Service {  
    MediaPlayer player; // 음악 재생을 위한 객체  
    int size;  
    NotificationManager manager;  
    private final int TEST_FLAG = 111;  
  
    @Override  
    public IBinder onBind(Intent intent) {  
        return null;  
    }  
  
    @Override  
    public void onCreate() {  
        manager = (NotificationManager)  
            getSystemService(Context.NOTIFICATION_SERVICE);  
        player = MediaPlayer.create(this, R.raw.davichi);  
        player.setLooping(true);  
    }  
}
```



# Start Service 예제 3(음악 연주)

## ■ onCreate( ), onDestroy( ), onStartCommand( ) 메소드

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
```

```
@Override
```

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    String title = intent.getStringExtra("title");  
    Toast.makeText(this, "Music Service가 시작되었습니다.\n 노래제목 : "  
        +title, Toast.LENGTH_LONG).show();  
  
    manager.notify(TEST_FLAG, makeNotification());  
    size = player.getDuration();  
    player.start();  
    MainActivity.bar.setMax(size);  
    MainActivity.textView.setText(getTime(player.getCurrentPosition())+ "/"  
        + getTime(size));  
  
    MyThread thread = new MyThread(player, size);  
    thread.start();  
    return START_STICKY;  
}
```



# Start Service 예제 3(음악 연주)

■ onCreate( ), onDestroy( ), onStartCommand( ) 메소드

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
private Notification makeNotification() {
    Notification.Builder builder = new Notification.Builder(this);
    Bitmap largelcon = BitmapFactory.decodeResource(getResources(),
                                                    R.mipmap.icon);

    builder.setTitle("서비스 안내")
        .setContentInfo("Content Info")
        .setContentText("음악 서비스가 시작되었습니다")
        .setSmallIcon(R.drawable.goldcoin)
        .setLargeIcon(largelcon);
    return builder.build();
}
```



# Start Service 예제 3(음악 연주)

■ onCreate( ), onDestroy( ), onStartCommand( ) 메소드

@Override

```
public void onDestroy() {  
    manager.cancelAll();  
    MainActivity.bar.setProgress(0);  
    player.stop();  
    Toast.makeText(this, "Music Service가 중지되었습니다",  
                   Toast.LENGTH_LONG).show();  
}
```

```
static public String getTime(int time) {  
    int second = time / 1000;  
    int minute = second / 60;  
    second %= 60;  
    return String.format(Locale.KOREA, "%02d:%02d", minute, second);  
}  
}
```



# Start Service 예제 3(음악 연주)

## ■ MediaPlayer 클래스

- MediaPlayer는 음원뿐만 아니라 영상 재생에도 사용되는 클래스
- 영상을 재생할 때는 화면 출력이 되어야 하므로 SurfaceView를 준비해야 하지만, 음원 재생은 특별한 뷰 없이 MediaPlayer만으로 간단하게 재생할 수 있음
- 음원을 재생할 때 재생이나 정지 버튼과 프로그래스바 등을 출력해야 한다면 별도의 뷰를 이용해 개발자가 직접 준비해야 함



# Start Service 예제 3(음악 연주)

## ■ onStartCommand()

- 다른 컴포넌트에서 startService()를 호출하여 서비스가 시작되면 이 메소드가 호출
- 작업이 완료되었을 경우 stopSelf()나 stopService()를 호출하여 서비스를 종료
- 그렇지 않으면 무한 실행

onCreate() → onStartCommand()  
→ Service Running → onDestroy()



# Start Service 예제 3(음악 연주)

- 메인 Java 코드 완성(MainActivity.java)
  - MusicService 클래스를 적용할 Intent 변수 1개와 Button 변수 1개를 전역 변수로 선언
  - 인텐트 변수를 생성하면서 MusicService 클래스를 생성자에게 넘김
  - activity\_main.xml의 버튼 1개를 버튼 변수에 적용시킴
  - <시작>을 클릭하면 startService( )를 호출하고 로그를 남김





# Start Service 예제 3(음악 연주)

## ■ 메인 Java 코드 완성

```
public class MainActivity extends AppCompatActivity {  
    static boolean isPlaying = false;  
    static ProgressBar bar;  
    static TextView textView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        final Intent intent = new Intent(getApplicationContext(), MusicService.class);  
        bar = findViewById(R.id.progress);  
        textView = findViewById(R.id.status);  
    }  
}
```



# Start Service 예제 3(음악 연주)

## ■ 메인 Java 코드 완성

```
final Button button = findViewById(R.id.start);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (!isPlaying) {
            isPlaying = true;
            intent.putExtra("title", "헤어졌다 만났다");
            startService(intent);
            button.setText("Music Service 중지");
        } else {
            stopService(intent);
            isPlaying = false;
            button.setText("Music Service 시작");
        }
    }
});
}
```



# Start Service 예제 3(음악 연주)

## ■ 메인 Java 코드 완성

```
@SuppressWarnings("HandlerLeak")
static Handler handler = new Handler() {
    @Override
    public void handleMessage(@NonNull Message msg) {
        if (msg.what == 0) {
            Bundle bundle = msg.getData();
            bar.setProgress(bundle.getInt("current"));
            textView.setText(bundle.getString("value"));
        }
    }
};
```



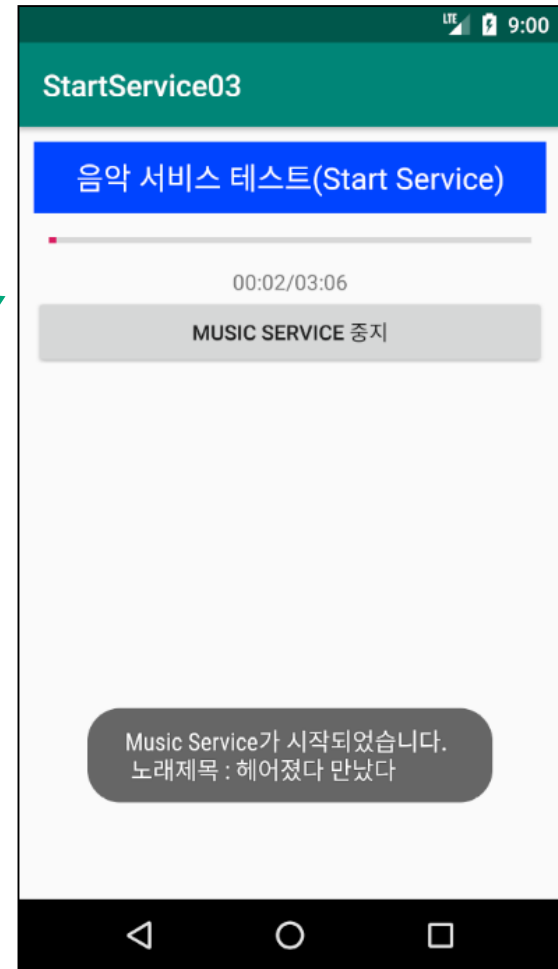
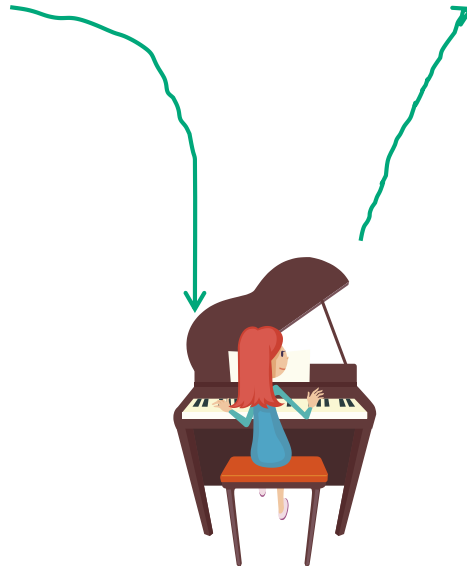
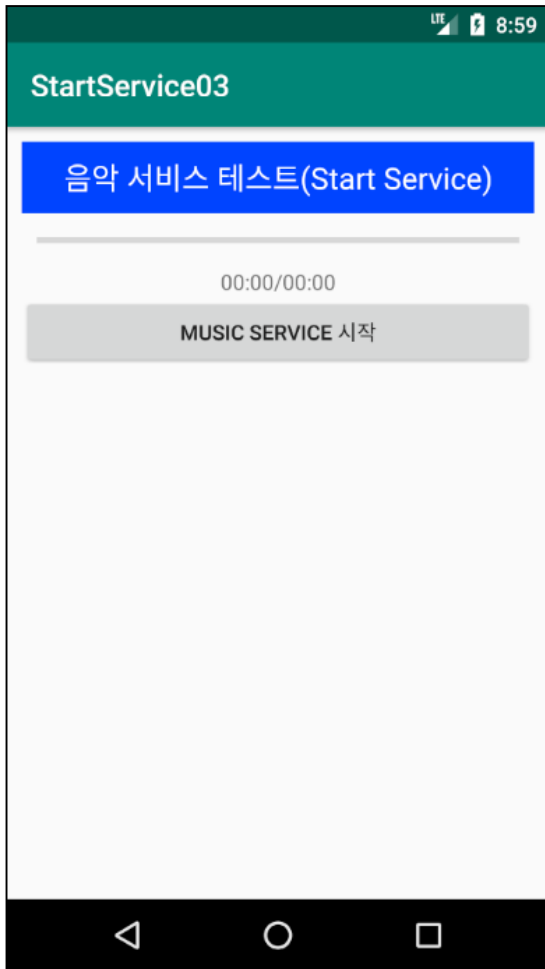
# Start Service 예제 3(음악 연주)

- Manifest.xml을 열고 <application>안에 서비스 등록
- <application> 안에 넣어주고, name 프러퍼티는 클래스 이름을 등록

```
.....  
<activity android:name="com.example.it_.musicservice.MainActivity">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>  
    <service  
        android:enabled="true"  
        android:name=".MusicService">  
    </service>  
</application>  
</manifest>
```

# Start Service 예제 3(음악 연주)

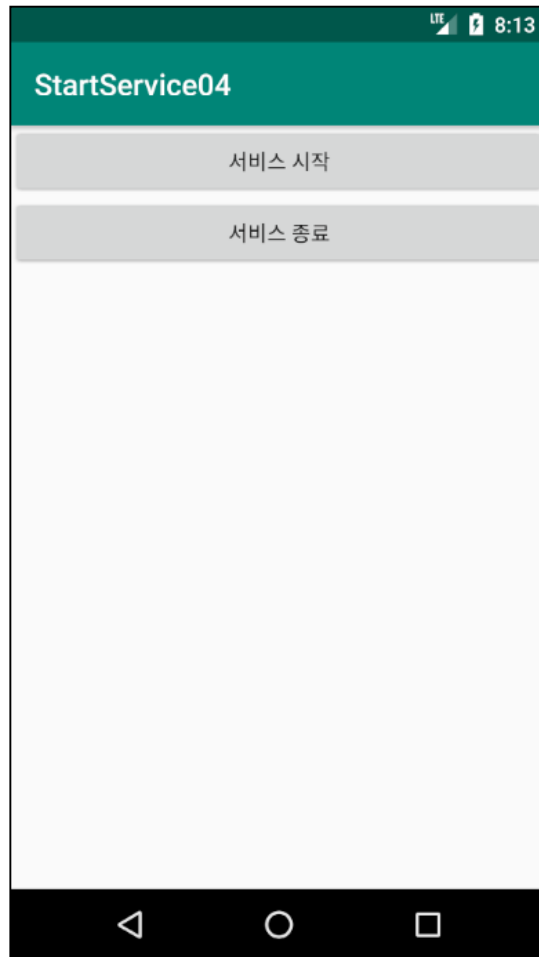
## ■ 실행 결과





# Start Service 예제 4

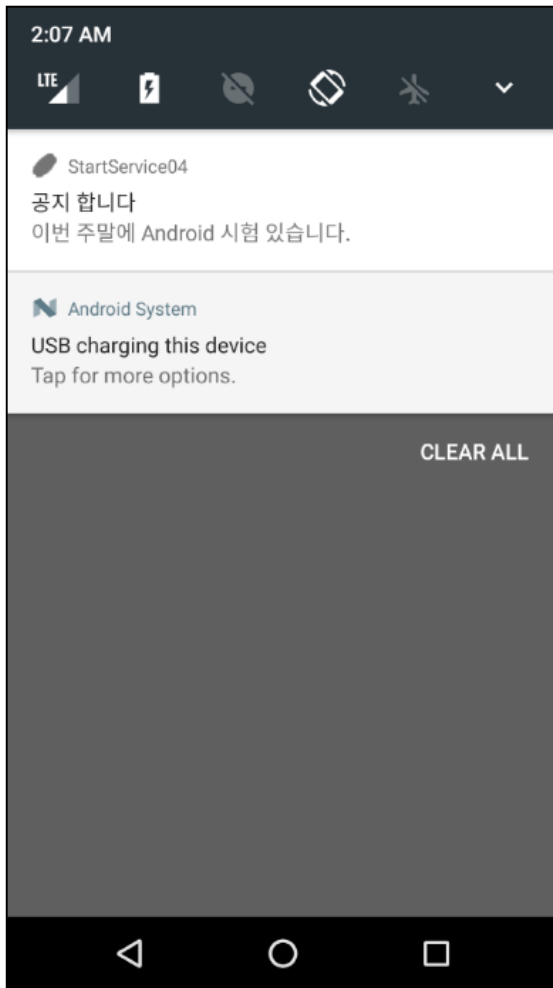
- 다음과 같은 Start Service를 구현하여보자





# Start Service 예제 4

## ■ 실행 화면





# Start Service 예제 4



## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/start"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="서비스 시작"/>
    <Button
        android:id="@+id/stop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="서비스 종료"/>
</LinearLayout>
```





# Start Service 예제 4

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    static Button button1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        final Intent intent = new Intent(MainActivity.this, MyService.class);  
  
        button1 = findViewById(R.id.start);  
        button1.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                startService(intent);  
            }  
        });  
    }  
};
```



# Start Service 예제 4

## ■ MainActivity.JAVA

```
Button button2 = findViewById(R.id.stop);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        stopService(intent);
    }
});
}
```



# Start Service 예제 4

## ■ MyService.JAVA

```
public class MyService extends Service {  
    NotificationManager manager;  
    ServiceThread thread;  
    Notification notification;  
  
    @Override  
    public IBinder onBind(Intent intent) {  
        return null;  
    }  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
    }  
}
```



# Start Service 예제 4



@Override

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    MainActivity.button1.setEnabled(false);  
    Toast.makeText(getBaseContext(), "Service 시작",  
                   Toast.LENGTH_SHORT).show();  
    manager = (NotificationManager) getSystemService(  
        Context.NOTIFICATION_SERVICE);  
    myServiceHandler handler = new myServiceHandler();  
    thread = new ServiceThread(handler);  
    thread.start();  
    return START_STICKY;  
}
```

@Override

```
public void onDestroy() {  
    Toast.makeText(getBaseContext(), "Service 끝",  
                   Toast.LENGTH_SHORT).show();  
    MainActivity.button1.setEnabled(true);  
    thread.stopForever();  
    manager.cancelAll();  
}
```



# Start Service 예제 4

## ■ MyService.JAVA

```
@SuppressWarnings("HandlerLeak")
class myServiceHandler extends Handler {
    @RequiresApi(api = Build.VERSION_CODES.JELLY_BEAM)
    @Override
    public void handleMessage(android.os.Message msg) {
        Intent intent = new Intent(MyService.this, MainActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(
            MyService.this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
        notification = new Notification.Builder(getApplicationContext())
            .setContentTitle("공지 합니다")
            .setContentText("이번 주말에 Android 시험 있습니다.")
            .setSmallIcon(R.drawable.goldcoin)
            .setTicker("알림!!!")
            .setContentIntent(pendingIntent)
            .build();
    }
}
```



# Start Service 예제 4

## ■ MyService.JAVA

```
notification.defaults = Notification.DEFAULT_SOUND;
notification.flags = Notification.FLAG_ONLY_ALERT_ONCE;
notification.flags = Notification.FLAG_AUTO_CANCEL;
manager.notify( 777 , notification);
Toast.makeText(MyService.this, "알림 도착",
               Toast.LENGTH_LONG).show();
    }
};
}
```



# Start Service 예제 4

## ■ ServiceThread.JAVA

```
public class ServiceThread extends Thread{  
    private Handler handler;  
    private boolean isRun = true;  
  
    public ServiceThread(Handler handler){  
        this.handler = handler;  
    }  
  
    public void stopForever(){  
        isRun = false;  
    }  
}
```



# Start Service 예제 4

## ■ ServiceThread.JAVA

```
public void run(){ //반복적으로 수행할 작업
    while(isRun){
        handler.sendMessage(0);
        try{
            Thread.sleep(10000); //10초씩 쉰다.
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```





# Start Service 예제 4



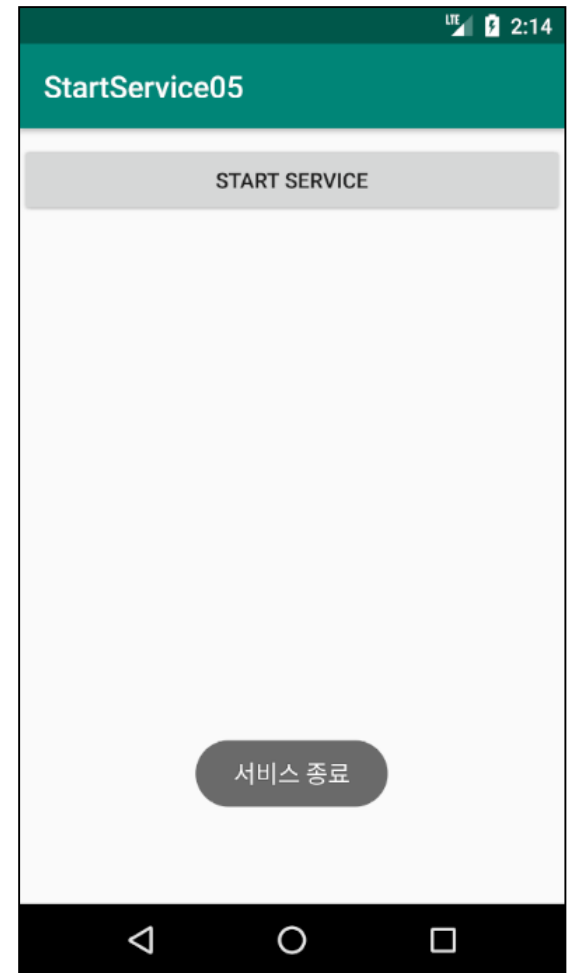
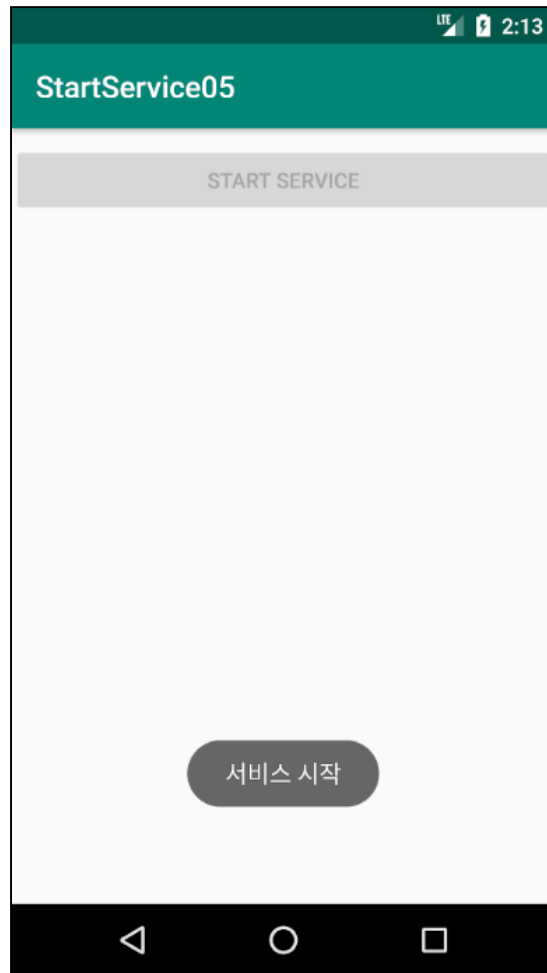
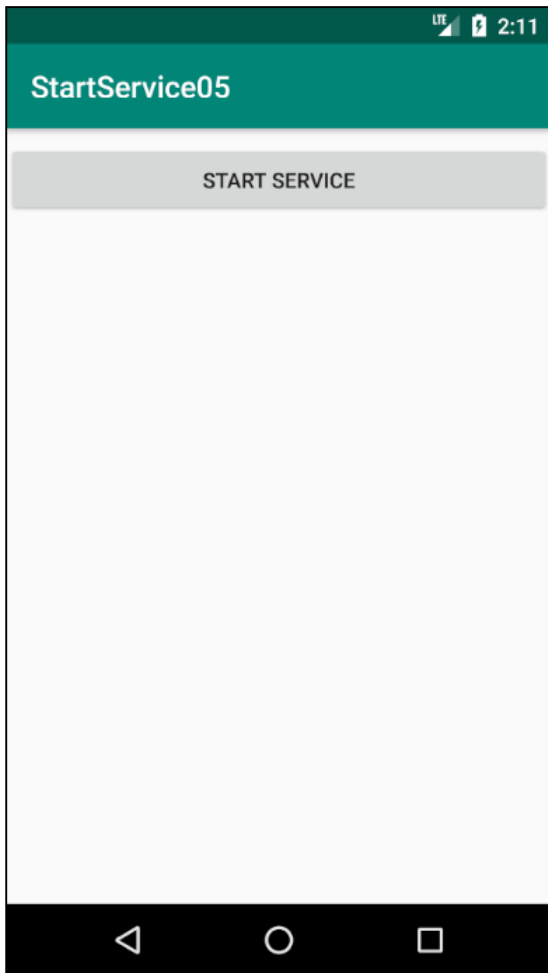
## ■ Manifest.xml

```
.....  
<service  
    android:name="com.example.startservice06.MyService"  
    android:enabled="true"  
    android:exported="true"></service>  
  
<activity android:name="com.example.startservice06.MainActivity">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
    </activity>  
</application>
```



# Start Service 예제 5

■ 다음과 같은 Start Service를 구현하여보자





# Start Service 예제 5

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Start Service"/>

</LinearLayout>
```



# Start Service 예제 5

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    static Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, MyService.class);
                startService(intent);
            }
        });
    }
}
```



# Start Service 예제 5

## ■ MyService.JAVA

```
public class MyService extends Service {  
  
    @Override  
    public IBinder onBind(Intent intent) {  
        return null;  
    }  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
    }  
  
    @Override  
    public int onStartCommand(Intent intent, int flags, int startId) {  
        final int id = startId;  
        MainActivity.button.setEnabled(false);  
        Toast.makeText(getApplicationContext(), "서비스 시작",  
                                Toast.LENGTH_SHORT).show();  
    }  
}
```



# Start Service 예제 5

## ■ MyService.JAVA

```
Runnable r = new Runnable() {  
    public void run() {  
        for (int i = 0; i < 3; i++) {  
            long endTime = System.currentTimeMillis() + 10*1000;  
            while (System.currentTimeMillis() < endTime) {  
                synchronized (this) {  
                    try {  
                        wait(endTime - System.currentTimeMillis());  
                    } catch (Exception e) {  
                    }  
                }  
            }  
        }  
        stopSelf(id);  
    }  
};
```



# Start Service 예제 5

## ■ MyService.JAVA

```
Thread t = new Thread(r);  
t.start();  
Toast.makeText(getApplicationContext(), "서비스중 입니다",  
                                Toast.LENGTH_SHORT).show();  
  
return START_STICKY;  
}
```

@Override

```
public void onDestroy() {  
    MainActivity.button.setEnabled(true);  
    Toast.makeText(getApplicationContext(), "서비스 종료",  
                                Toast.LENGTH_SHORT).show();  
}  
}
```



# Start Service 예제 5



## ■ AndroidManifest.xml

```
.....
<activity android:name="com.example.startservice05.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<service android:name="com.example.startservice05.MyService"/>
</application>

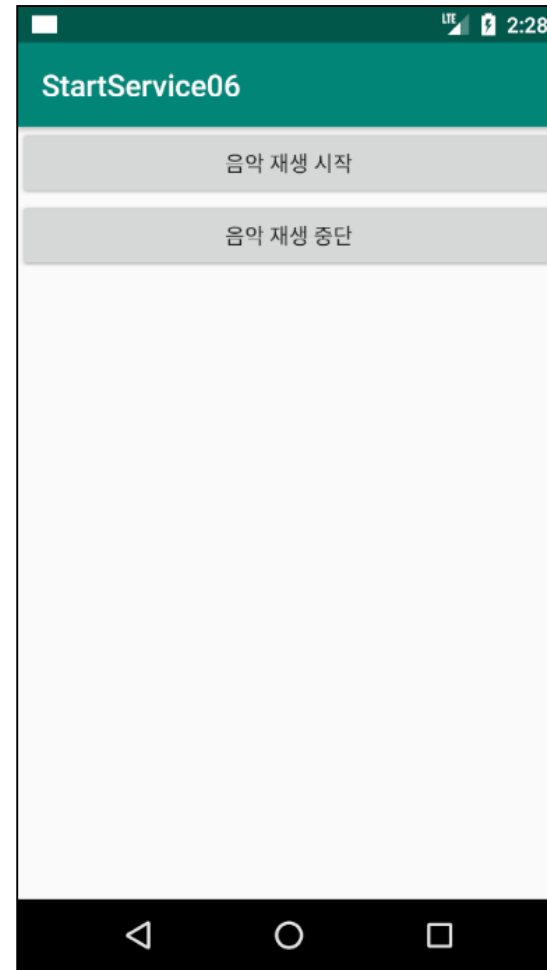
</manifest>
```





# Start Service 예제 6

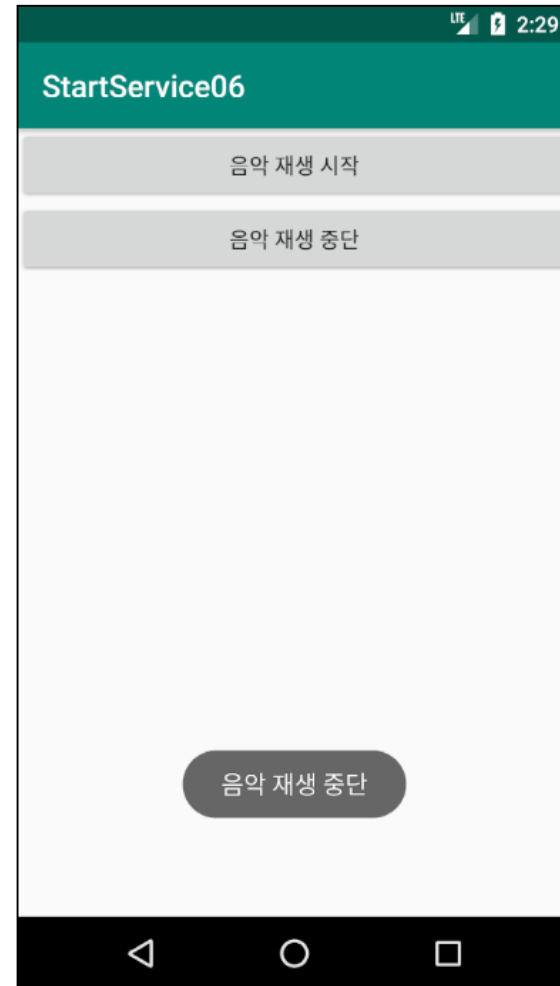
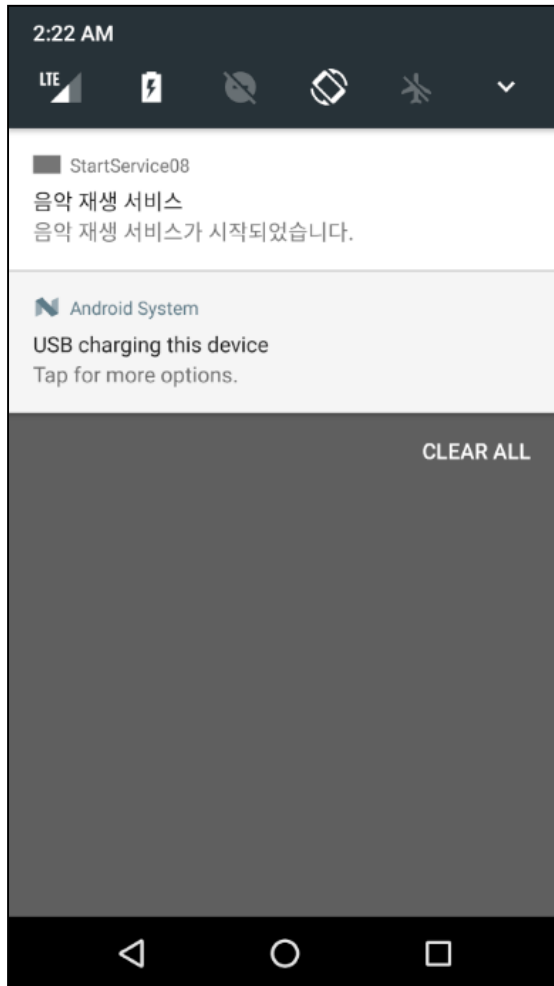
- StartService를 이용하여 음악을 재생하여 보자





# Start Service 예제 6

## ■ 실행 화면





# Start Service 예제 6

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/startBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/startStr" />
    <Button
        android:id="@+id/stopBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/stopStr" />
</LinearLayout>
```



# Start Service 예제 6

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity
                                implements View.OnClickListener {

    Button[] buttons;
    int[] buttonLables = {R.id.start, R.id.stop};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        buttons = new Button[buttonLables.length];

        for (int i = 0; i < buttonLables.length; i++) {
            buttons[i] = findViewById(buttonLables[i]);
            buttons[i].setOnClickListener(this);
        }
    }
}
```



# Start Service 예제 6

## ■ MainActivity.JAVA

@Override

```
public void onClick(View view) {  
    Intent intent = new Intent(this, MyService.class);  
    if (view == buttons[0])    //[음악 재생 시작]버튼 클릭시  
        startService(intent);  
    else if (view == buttons[1])    //[음악 재생 중단]버튼 클릭시  
        stopService(intent);  
}  
}
```



# Start Service 예제 6

## ■ MyService.JAVA

```
public class MyService extends Service {
    NotificationManager manager;
    MediaPlayer player;
    private int NOTIFICATION_ID = 0;

    public void onCreate(){
        manager = (NotificationManager)
            getSystemService(NOTIFICATION_SERVICE);
        try{
            player = MediaPlayer.create(this, R.raw.davichi);
            player.prepare();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```



# Start Service 예제 6

## ■ MyService.JAVA

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAM)
public void onStart(Intent intent, int startId ){
    showNotification();
    try{
        player.start();
    }catch(Exception e){
        e.printStackTrace();
    }
}

public void onDestroy(){
    manager.cancelAll();
    player.stop(); //음악재생중단
    player.release(); //음악재생을 해제
    player = null; //리소스 반환

    Toast.makeText(this, R.string.stopStr, Toast.LENGTH_SHORT).show();
}
```



# Start Service 예제 6

## ■ MyService.JAVA

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAM)
private void showNotification(){
    CharSequence title = getText(R.string.title);
    CharSequence message = getText(R.string.message);

    Intent intent = new Intent(this, MyService.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
        intent, 0);

    Notification.Builder builder = new Notification.Builder(this);
    builder.setContentTitle(title)
        .setSmallIcon(R.drawable.cloudicon)
        .setContentText(message)
        .setContentIntent(pendingIntent);
    manager.notify(NOFITICATION_ID, builder.build());
}
```





# Start Service 예제 6

## ■ MyService.JAVA

```
@Override  
public IBinder onBind(Intent arg0) {  
    return null;  
}  
}
```



# Start Service 예제 6

## ■ strings.xml

```
<resources>
  <string name="app_name">StartService06</string>

  <string name="startStr">음악 재생 시작</string>
  <string name="stopStr">음악 재생 중단</string>
  <string name="ticker">음악이 재생됩니다.</string>
  <string name="title">음악 재생 서비스</string>
  <string name="message">음악 재생 서비스가 시작되었습니다.</string>

</resources>
```



# Start Service 예제 6

## ■ 메니페스트

```
<activity android:name="com.example.startservice06.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

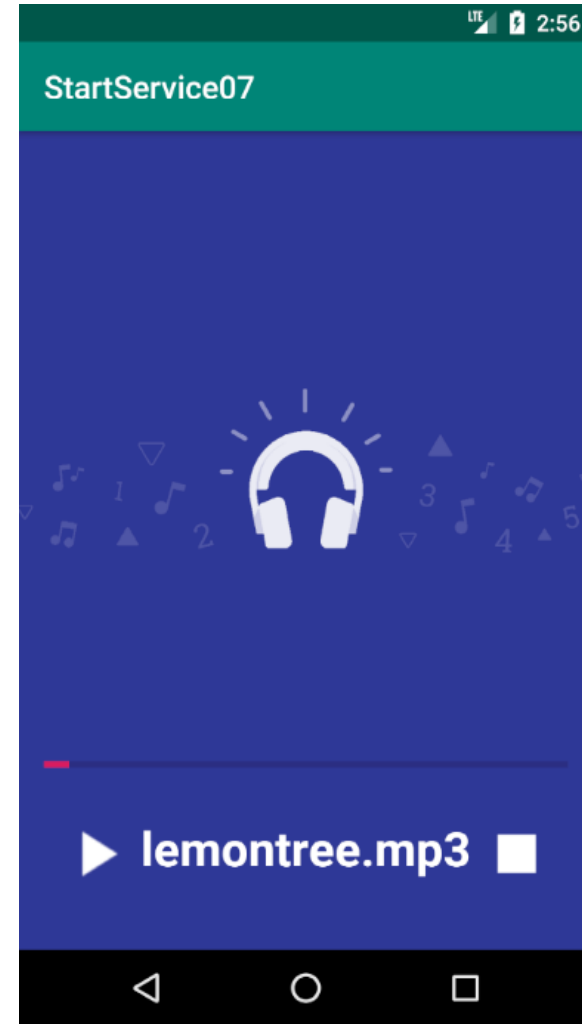
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<service android:name="com.example.startservice06.MyService"></service>
</application>

</manifest>
```



# Start Service 예제 7

- 다음과 같은 StartService를 만들어 보자





# Start Service 예제 7

## ■ 음원 준비

Device File Explorer

Emulator Nexus\_5\_API\_24 Android 7.0, API 24

Name	Permissio...	Date	Size
▶ acct	drwxr-xr-x	2020-07-07 00:37	0 B
▶ cache	drwxrwx---	2020-07-07 00:37	4 KB
▶ config	drwxr-xr-x	2020-07-07 00:37	0 B
▶ d	lrwxrwxrwx	1970-01-01 00:00	17 B
▶ data	drwxrwx--x	2020-07-07 00:37	4 KB
▶ dev	drwxr-xr-x	2020-07-07 00:37	2.4 KB
▶ etc	lrwxrwxrwx	1970-01-01 00:00	11 B
▼ mnt	drwxr-xr-x	2020-07-07 00:37	220 B
▶ appfuse	drwx--x--x	2020-07-07 00:37	40 B
▶ expand	drwxrwx--x	2020-07-07 00:37	40 B
▶ obb	drwxr-xr-x	2020-07-07 00:37	40 B
▶ runtime	drwx-----	2020-07-07 00:37	100 B
▼ sdcard	lrwxrwxrwx	2020-07-07 00:37	21 B
▶ Alarms	drwxrwx--x	2020-07-07 00:38	4 KB
▶ Android	drwxrwx--x	2020-07-07 00:38	4 KB
▶ DCIM	drwxrwx--x	2020-07-07 00:38	4 KB
▶ Download	drwxrwx--x	2020-07-07 00:38	4 KB
▶ Movies	drwxrwx--x	2020-07-07 00:38	4 KB
▼ Music	drwxrwx--x	2020-07-07 00:38	4 KB
lemontree.mp3	-rw-rw----	2020-07-08 02:51	4.4 MB
▶ Notifications	drwxrwx--x	2020-07-07 00:38	4 KB
▶ Pictures	drwxrwx--x	2020-07-07 00:38	4 KB
▶ Podcasts	drwxrwx--x	2020-07-07 00:38	4 KB
▶ Ringtones	drwxrwx--x	2020-07-07 00:38	4 KB
▶ secure	drwx-----	2020-07-07 00:37	60 B
▶ user	drwxr-xr-x	2020-07-07 00:37	60 B
▶ oem	drwxr-xr-x	1970-01-01 00:00	0 B
▶ proc	dr-xr-xr-x	2020-07-07 00:37	0 B
▶ root	drwx-----	2017-05-19 19:49	0 B
▶ sbin	drwxr-x---	1970-01-01 00:00	0 B
▶ sdcard	lrwxrwxrwx	1970-01-01 00:00	21 B



# Start Service 예제 7

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#303897"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/background"
        android:layout_alignParentTop="true" />
```



# Start Service 예제 7

## ■ 사용자 인터페이스

```
<ImageView  
    android:id="@+id/lab1_play"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/ic_play"  
    android:layout_alignParentBottom="true"  
    android:layout_marginBottom="36dp"  
    android:layout_marginLeft="24dp"  
    android:clickable="true"/>
```



# Start Service 예제 7

## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/lab1_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@android:color/white"
    android:textStyle="bold"
    android:textSize="30dp"
    android:layout_alignParentBottom="true"
    android:layout_alignTop="@id/lab1_play"
    android:layout_marginLeft="16dp"
    android:layout_centerHorizontal="true"
    android:layout_centerInParent="true"
    android:layout_centerVertical="true" />
```





# Start Service 예제 7

```
<ImageView
```

```
    android:id="@+id/lab1_stop"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/ic_stop"  
    android:layout_alignParentBottom="true"  
    android:layout_alignParentRight="true"  
    android:layout_marginBottom="36dp"  
    android:layout_marginRight="24dp"  
    android:clickable="true"/>
```

```
<ProgressBar
```

```
    android:id="@+id/lab1_progress"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    style="@style/Base.Widget.AppCompat.ProgressBar.Horizontal"  
    android:layout_above="@id/lab1_title"  
    android:layout_marginBottom="24dp"  
    android:layout_marginLeft="16dp"  
    android:layout_marginRight="16dp"/>
```

```
</RelativeLayout>
```



# Start Service 예제 7

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity
                                implements View.OnClickListener {

    ImageView play, stop;
    ProgressBar progressBar;
    boolean runThread;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        progressBar = findViewById(R.id.lab1_progress);
        TextView textView = findViewById(R.id.lab1_title);
        textView.setText("lemontree.mp3");
        play = findViewById(R.id.lab1_play);
        stop = findViewById(R.id.lab1_stop);
        play.setOnClickListener(this);
        stop.setOnClickListener(this);
        stop.setEnabled(false);
    }
}
```



# Start Service 예제 7

## ■ MainActivity.JAVA

```
String filePath = Environment.getExternalStorageDirectory().  
    getAbsolutePath() + "/Music/lemontree.mp3";  
if (ContextCompat.checkSelfPermission(this, Manifest.permission.  
    READ_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {  
    ActivityCompat.requestPermissions(this,  
        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 100);  
}  
registerReceiver(receiver, new  
    IntentFilter("com.example.PLAY_TO_ACTIVITY"));  
Intent intent = new Intent(this, PlayService.class);  
intent.putExtra("filePath", filePath);  
startService(intent);  
}
```



# Start Service 예제 7

## ■ MainActivity.JAVA

```
class ProgressThread extends Thread {  
    @Override  
    public void run() {  
        while (runThread) {  
            progressBar.incrementProgressBy(1000);  
            SystemClock.sleep(1000);  
            if (progressBar.getProgress() == progressBar.getMax()) {  
                runThread = false;  
            }  
        }  
    }  
}
```



# Start Service 예제 7

## ■ MainActivity.JAVA

```
BroadcastReceiver receiver = new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        String mode = intent.getStringExtra("mode");  
        if (mode != null) {  
            if (mode.equals("start")) {  
                int duration = intent.getIntExtra("duration", 0);  
                progressBar.setMax(duration);  
                progressBar.setProgress(0);  
            } else if (mode.equals("stop")) {  
                runThread = false;  
            } else if (mode.equals("restart")) {  
                int duration = intent.getIntExtra("duration", 0);  
                int current = intent.getIntExtra("current", 0);  
                progressBar.setMax(duration);  
                progressBar.setProgress(current);  
                runThread = true;  
                ProgressThread thread = new ProgressThread();  
                thread.start();  
            }  
        }  
    }  
};
```



# Start Service 예제 7

## ■ MainActivity.JAVA

```
        play.setEnabled(false);
        stop.setEnabled(true);
    }
}
};
```

@Override

```
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(reciever);
}
```



# Start Service 예제 7

## ■ MainActivity.JAVA

@Override

```
public void onClick(View v) {  
    if (v == play) {  
        Intent intent = new Intent("com.example.PLAY_TO_SERVICE");  
        intent.putExtra("mode", "start");  
        sendBroadcast(intent);  
  
        runThread = true;  
        ProgressThread thread = new ProgressThread();  
        thread.start();  
        play.setEnabled(false);  
        stop.setEnabled(true);  
    } else if (v == stop) {
```



# Start Service 예제 7

## ■ MainActivity.JAVA

```
    } else if (v == stop) {  
        Intent intent = new Intent("com.example.PLAY_TO_SERVICE");  
        intent.putExtra("mode", "stop");  
        sendBroadcast(intent);  
        runThread = false;  
        progressBar.setProgress(0);  
        play.setEnabled(true);  
        stop.setEnabled(false);  
    }  
}  
}
```





# Start Service 예제 7

## ■ PlayService.JAVA

```
public class PlayService extends Service
    implements MediaPlayer.OnCompletionListener{

    MediaPlayer player;
    String filePath;

    BroadcastReceiver receiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String mode=intent.getStringExtra("mode");
            if(mode != null){
                if(mode.equals("start")){
                    try{
                        if(player != null && player.isPlaying()){
                            player.stop();
                            player.release();
                            player=null;
                        }
                    }
                }
            }
        }
    }
```



# Start Service 예제 7

## ■ PlayService.JAVA

```
player=new MediaPlayer();  
player.setDataSource(filePath);  
player.prepare();  
player.start();  
  
Intent alntent = new Intent("com.example.PLAY_TO_ACTIVITY");  
alntent.putExtra("mode","start");  
alntent.putExtra("duration", player.getDuration());  
sendBroadcast(alntent);  
}catch (Exception e){  
    e.printStackTrace();  
}
```



# Start Service 예제 7

## ■ PlayService.JAVA

```
        }else if(mode.equals("stop")){  
            if(player != null && player.isPlaying()){  
                player.stop();  
                player.release();  
                player=null;  
            }  
        }  
    }  
};
```

### @Override

```
public void onCompletion(MediaPlayer mp) {  
    Intent intent=new Intent("com.example.PLAY_TO_ACTIVITY");  
    intent.putExtra("mode", "stop");  
    sendBroadcast(intent);  
    stopSelf();  
}
```



# Start Service 예제 7

## ■ PlayService.JAVA

@Override

```
public void onCreate() {  
    super.onCreate();  
    registerReceiver(receiver,  
        new IntentFilter("com.example.PLAY_TO_SERVICE"));  
}
```

@Override

```
public void onDestroy() {  
    super.onDestroy();  
    unregisterReceiver(receiver);  
}
```



# Start Service 예제 7

## ■ PlayService.JAVA

@Override

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    filePath=intent.getStringExtra("filePath");  
    if(player != null){  
        Intent alntent = new Intent("com.example.PLAY_TO_ACTIVITY");  
        alntent.putExtra("mode", "restart");  
        alntent.putExtra("duration", player.getDuration());  
        alntent.putExtra("current", player.getCurrentPosition());  
        sendBroadcast(alntent);  
    }  
    return super.onStartCommand(intent, flags, startId);  
}
```

@Override

```
public IBinder onBind(Intent intent) {  
    return null;  
}  
}
```



# Start Service 예제 7

## ■ 메니페스트

```
android:roundIcon="@mipmap/ic_launcher_round"
android:supportRtl="true"
android:theme="@style/AppTheme">
<service
    android:name="com.example.startservice07.PlayService"
    android:enabled="true"
    android:exported="true"></service>

<activity android:name="com.example.startservice07.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>
```