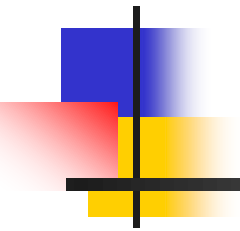
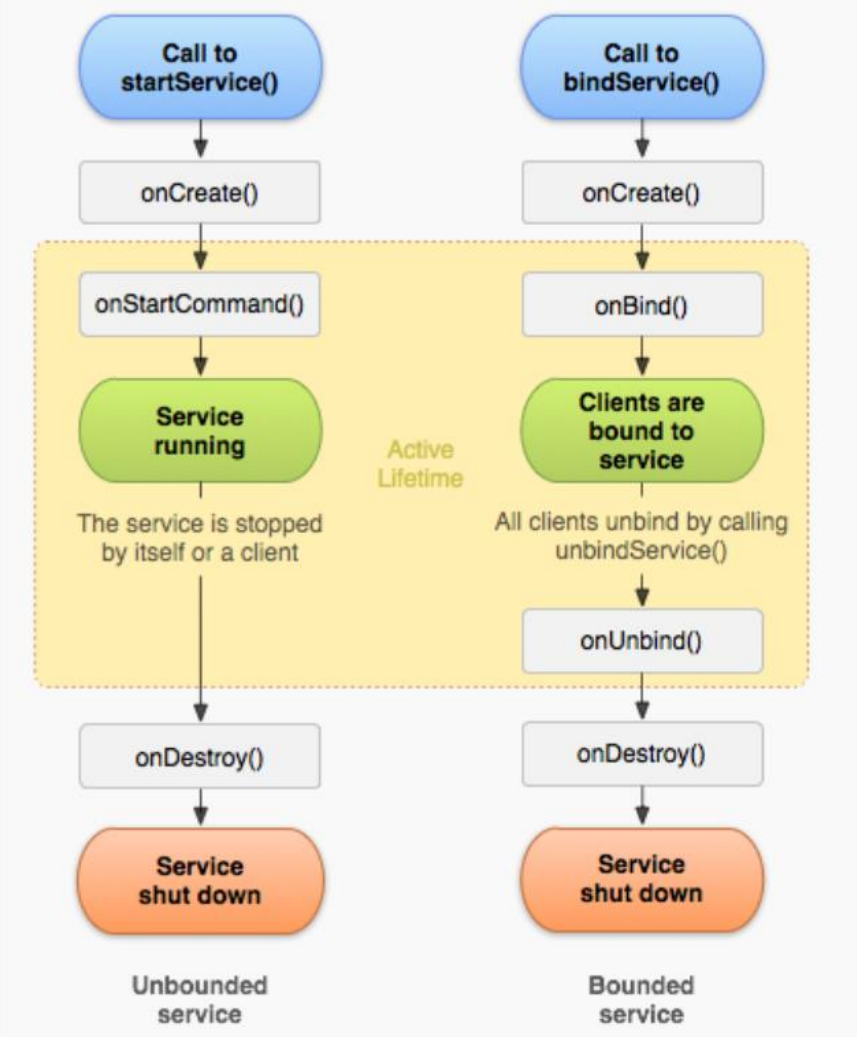


Bound Service 실습

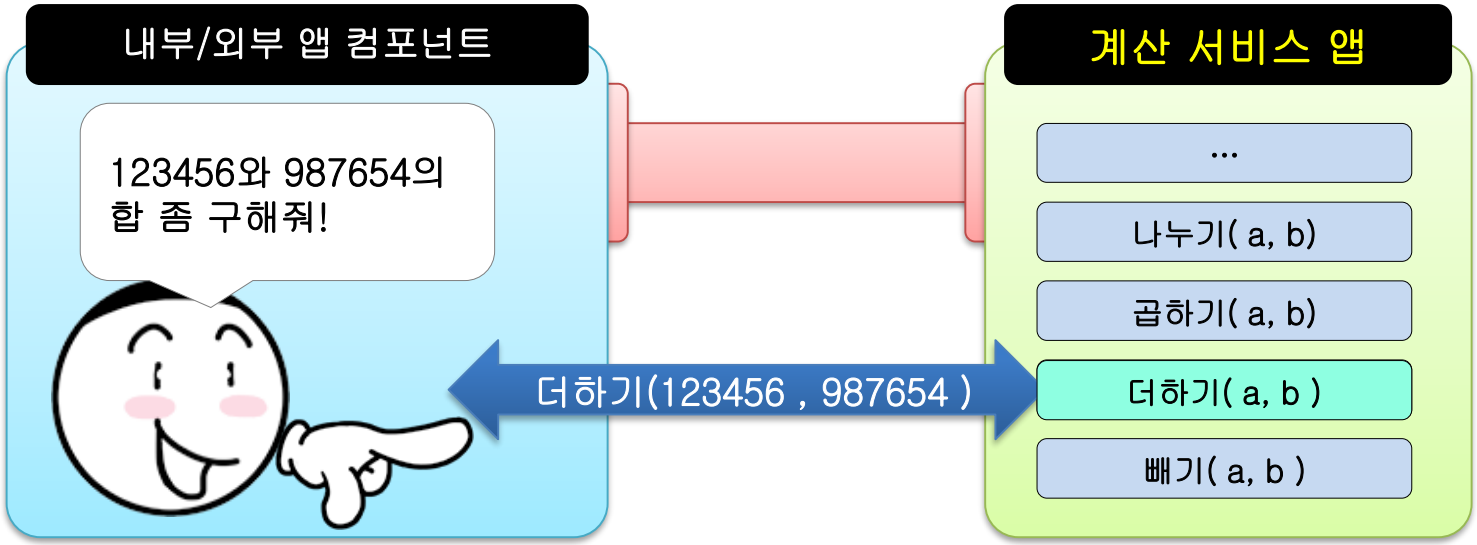
배 희호 교수
경복대학교
스마트IT과

Service



Bound Service

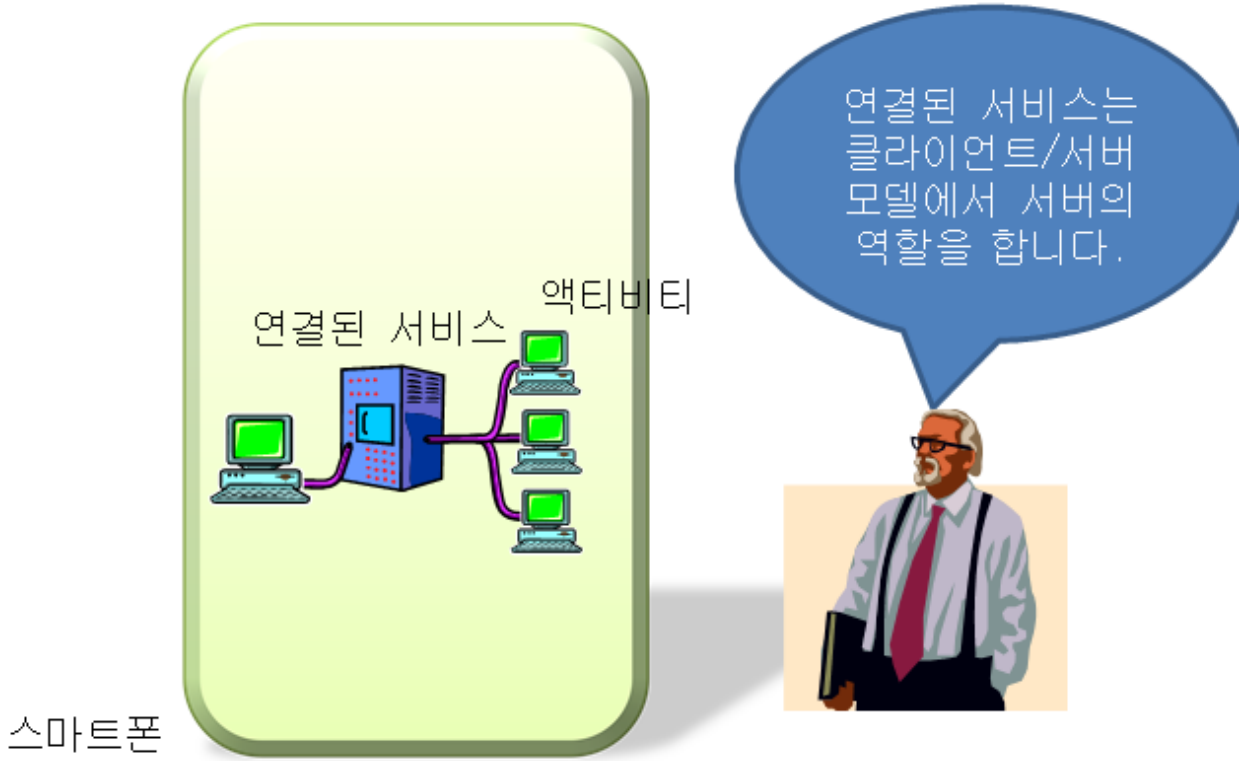
- Bound는 외부 Library를 사용하는 것과 매우 유사



- Service로 연결될 Component는 Service에 존재하는 함수들을 마치 Library를 가져다 사용하듯이 쓸 수 있음
- 내/외부 앱과 계산 Service의 관계는 서로 의존적이라 Service를 요청한 Component가 종료되면 Service 연결도 끊어짐. 반대로 Service는 내/외부 Component가 연결을 끊기 전까지 요청한 작업이 완료되었더라도 연결은 유지

Bound Service

- 이 Service는 마치 Client-Server와 같이 동작함
  - Service가 Server 역할을 함
  - 작업 결과를 호출자에게 전달할 수 있고 프로세스간 통신(IPC)를 수행할 수도 있음



Bound Service

- Binding된 구성 요소가 모두 Binding을 해제하면 Service는 소멸함
- Binding을 하려면 Service와의 연결을 모니터링하는 ServiceConnection의 구현을 해야 함
  - Android 시스템이 Client와 Service 사이의 연결을 생성하는 경우, 시스템은 onServiceConnected()를 ServiceConnection에서 호출하여 Client가 Service와 통신하는 데 사용할 수 있도록 IBinder를 전달함
- 호출 메소드 : bindService()
- 콜백 메소드 : onBind()
- 구현 해야하는 클래스 : ServiceConnetion

Bound Service

- IBinder 생성 방법
  - Binder 클래스 확장
    - 동일한 프로세스 내에서 Server-Client 형태로 상호작용 하는 경우 사용하는 방식 (대부분)
  - Messenger 사용
    - 여러 프로세스에서 적용되도록 할 때 사용하는 방식
    - IPC(프로세스간 통신)를 구현하는 가장 간단한 방법
  - AIDL 사용
    - 여러 프로세스에서 적용되어야 할 때 객체를 OS가 이해할 수 있는 원시 유형으로 해체한 다음 여러 프로세스에 걸쳐 마샬링하여 IPC를 수행하지만 대부분의 경우에는 바인드된 서비스를 사용하기 위해서 AIDL을 사용하지 않음

Bound Service

- 마샬링(Marshalling)
  - 객체의 메모리 구조를 저장이나 전송을 위해서 적당한 자료 형태로 변형하는 것을 의미
  - Marshalling은 보통 서로 다른 컴퓨터 혹은 서로 다른 프로그램 간에 데이터가 이동되어야 할 경우 사용
  - Marshalling을 수행함으로써 복잡한 통신, 사용자 정의/복잡한 구조의 객체들을 사용하는 대신, 단순한 primitive 들을 사용할 수 있음
  - Marshalling의 반대말은 Unmarshalling 이라고 함

Bound Service

- startService() 메소드 대신 bindService() 메소드를 통해 시작되는 Service를 서비스 바인딩(Service Bind 혹은 Bound Service)라고 함
- Activity는 Service에 어떠한 요청을 할 수 있고, Service로부터 어떠한 결과를 받을 수 있음
- 프로세스간 통신에도 사용 가능
- Service Binding은 연결된 Activity가 사라지면 Service도 소멸 됨 (즉 Background에서 무한히 실행되지 않음)
- 하나의 Service에 다수의 Activity 연결 가능
- 애플리케이션 안의 기능을 외부에 제공하는 경우에 많이 사용

Bound Service

- Binder
  - Linux Kernel에 있는 Binder를 통해 관리
  - Application과 Service들이 서로 다른 프로세스에서 실행 중인 경우, 프로세서간의 통신 및 데이터 공유의 관련 문제가 발생 -> 이때 IPC를 통해 프로세스간 통신 (필연적으로 IPC 통신의 경우, 시스템 오버헤드 및 보안 관련 문제를 야기할 수 있는 가능성이 항상 존재)
- Android Binder는 IPC 통신을 제공하는 프로토콜
  - 공유 메모리를 통한 성능 향상, 프로세서마다 요청 처리를 위한 스레드 풀
  - 오브젝트 맵핑과 레퍼런스 카운팅 기능 제공
  - 바인더를 통해 IPC가 이루어 질 때는 기본적으로는 동기 방식
  - 요청한 프로세서는 요청 결과를 얻을 때까지 대기 상태

Bound Service

- Binder
  - ServiceManager -> BinderDriver에 대한 special Node 0로 동작
  - ServiceProvider -> 자신의 RPC에 대한 Interface를 ServiceManager proxy object에 대한 Interface를 통해 등록
    - 이 과정을 통해 BinderDriver에서는 ServiceProvider로의 path를 생성
  - Service User -> 사용하려는 Service에 대한 Interface를 생성 위해 ServiceManager에게 RPC call을 통해 요청
  - ServiceManager -> Bind되어있는 Object들의 목록에서 해당하는 ServiceProvider에 대한 Node를 요청한 ServiceUser에게 return
  - ServiceUser -> return된 Node정보를 통해 해당 Service Provider로의 RPC를 수행

Bound Service

- Service에서 Binder를 사용하는 이유
  - Activity가 Service에 Binding되면, 그 Activity는 해당 서비스 인스턴스 자체에 대한 레퍼런스를 유지
  - Service가 Binding 된 후, Service가 가지고 있는 모든 public 메소드와 프로퍼티는 onServiceConnected 핸들러를 통해서 얻어진 serviceBinder 객체를 통해서 이용 가능
  - 일반 Service는 Client가 서비스의 동작을 알지 못하게 은밀히 동작 그러나 Binding된 Service는 Activity가 상태를 파악하고 제어
  - Broadcast 인텐트 또는 Service 시작 시 사용되는 인텐트 익스트라 번들을 이용해서 간단히 다른 프로세스로 실행 중인 서비스와 간단히 통신 가능
  - 좀더 단단히 결합된 연결을 원할 경우에는 AIDL 이용

Bound Service

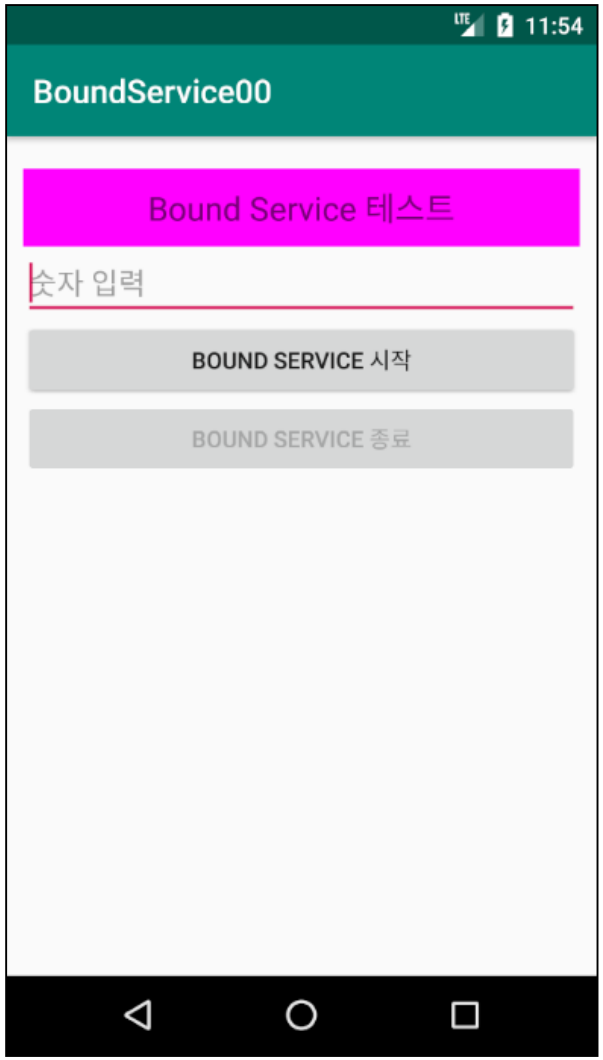
- startService
  - 받는 쪽과 묶어서 돌아가는 구조
  - 호출 순서 : onCreate() -> onStart() -> ... -> onDestory()
  - service가 실행중인 상태라면 startService()가 호출되어도 상태 유지, 서비스 종료상태에서 stopService()하면 아무 일도 발생하지 않음
  - bindService로 실행된 서비스를 startService로 호출하게 되면 이전(bindService) 관리는 무시되고 새로운 상태로 변경
  - service가 resource를 너무 많이 사용하게 되면 system에 의해 종료될 수 있음

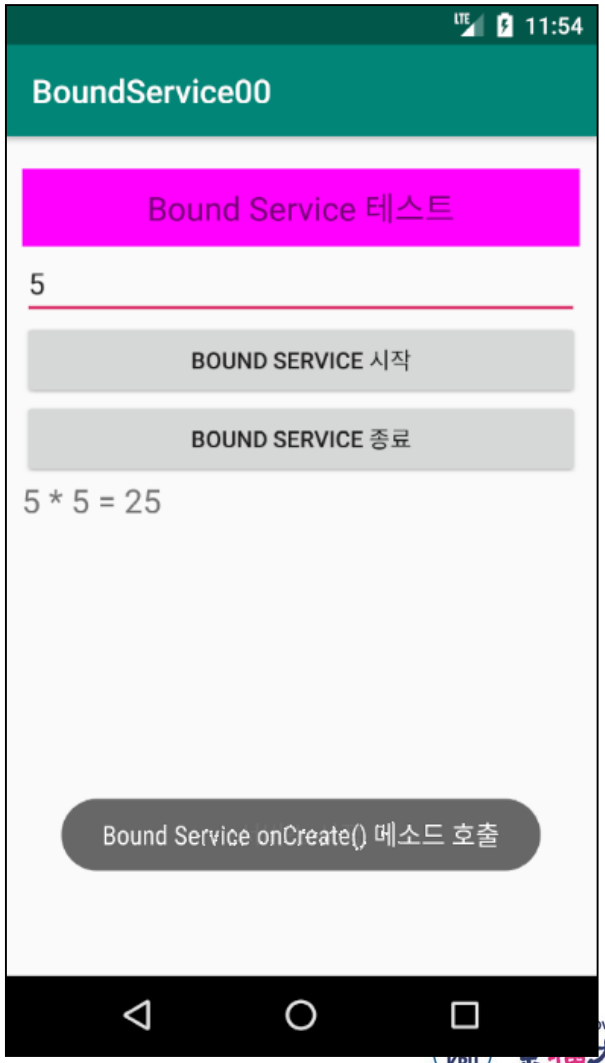
Bound Service

- bindService
  - 서비스 받는 쪽과 상관없이 돌아가는 구조
  - 호출 순서 : onCreate() -> onBind() -> ... -> onRebind -> ... -> onUnbind() -> onDestory()
  - 호출한 객체가 존재하면 system은 service가 필요하다고 판단하여 실행, 재실행 등으로 유지
  - 호출 객체 소멸 시 system에서 필요없다고 판단하면 언제든지 중지

Bound Service 예제 0

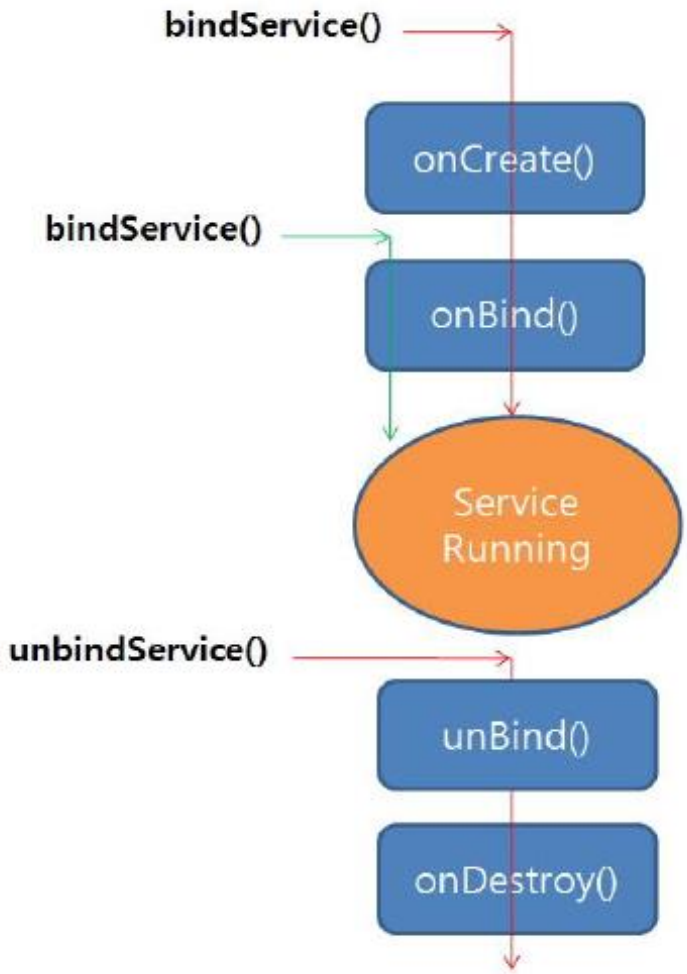
- Bound Service를 만들어보자





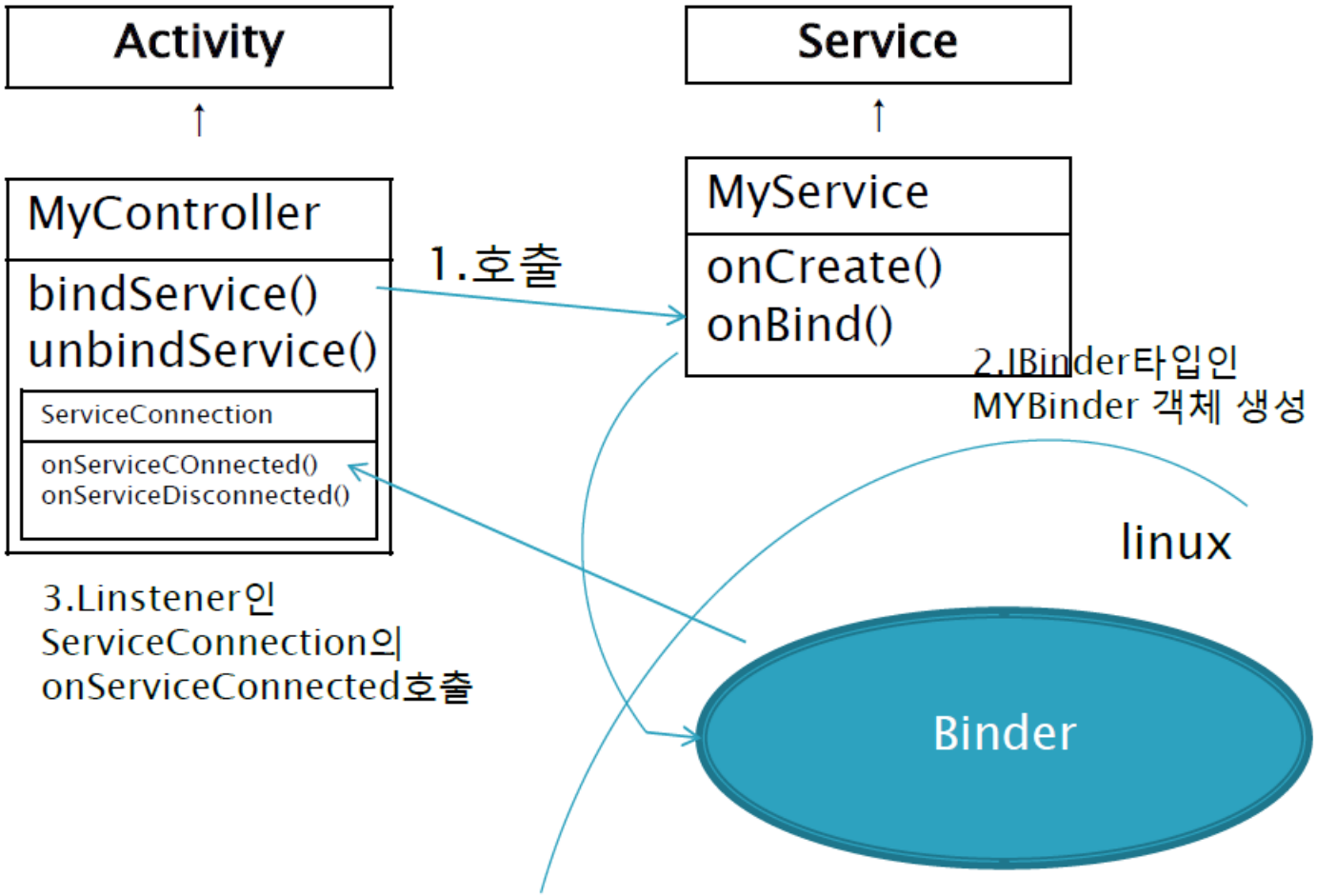
Bound Service 예제 0

- bindService( ) 메소드 이용 시 생명주기



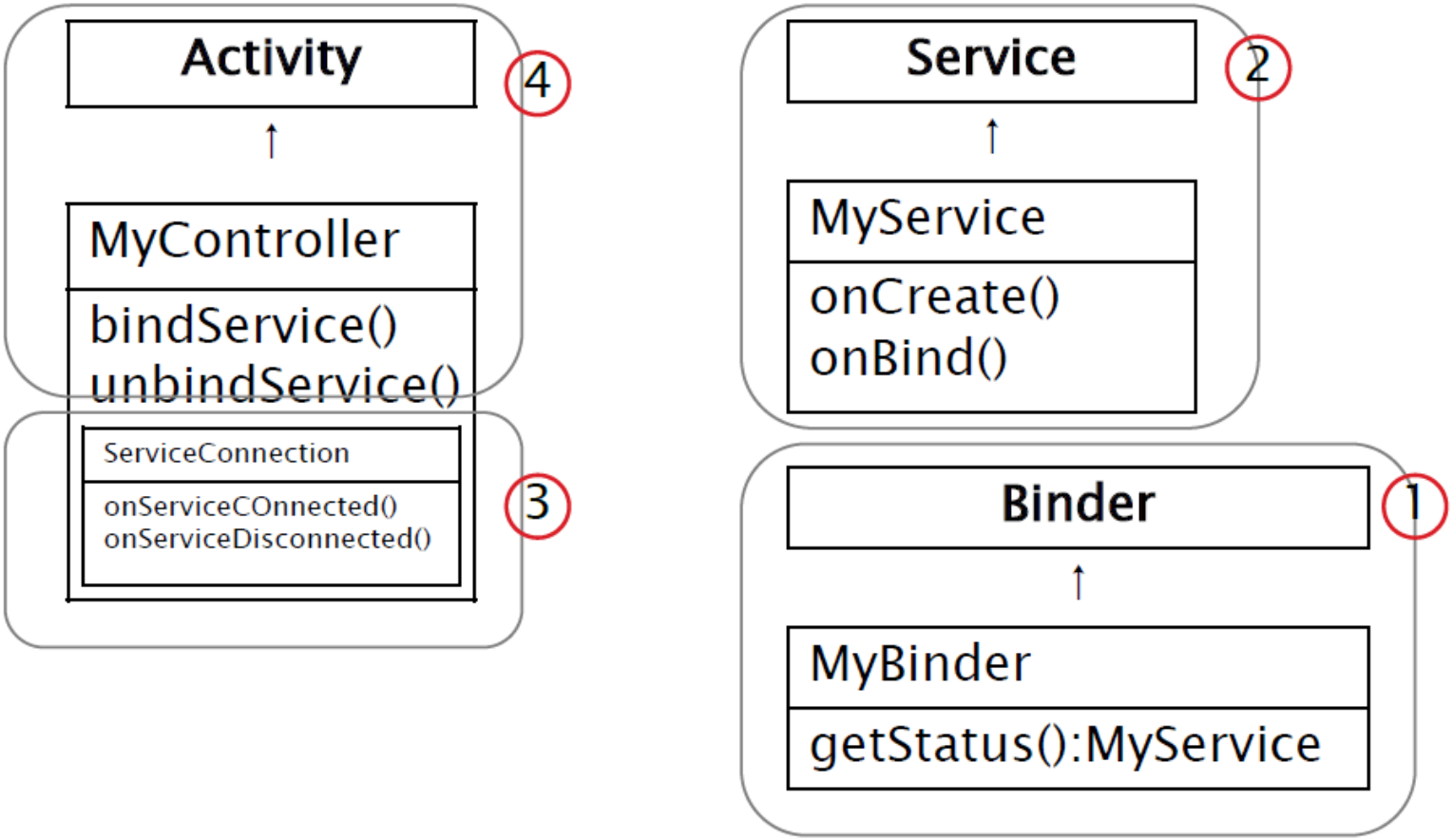
Bound Service 예제 0

- 서비스 동작



Bound Service 예제 0

- 서비스 작성 순서



Bound Service 예제 0

- Android Component는 bindService() 메소드를 호출함으로써 Bound Service에 결속(Binding) 할 수 있음
- Bound Service를 사용하는 경우
  - Activity나 다른 Component를 이용해서 Service와 상호 연동하고자 할 때
  - 나의 App의 일부 기능을 IPC를 이용해서 다른 App으로 하여금 이용할 수 있도록 하기 위해
- Bound Service를 이용할 수 있게 하려면, Service의 인터페이스를 정의한 IBinder를 반환하는 onBind() 콜백 메소드를 구현해야만 함
- Service를 이용하고자 하는 Component는 bindService() 메소드를 통해서 Service가 제공하는 Interface를 전달 받을 수 있으며, Service에서 제공하는 Interface 기능을 사용할 수 있음

Bound Service 예제 0

- Bound Service는 자신에게 결속(Binding)된 Component가 하나라도 존재하면 계속 작동 가능한 상태에 있음
- 그렇기 때문에 Service에 연결된 Component가 하나라도 없으면, 시스템에서는 Bound 서비스를 종료함
- Bound 서비스만 이용했다면, stopSelf()나 stopService()와 같은 함수로 종료시키지 않아도 됨
- Bound 서비스를 생성하기 위해서, 가장 먼저 해야 하는 것은 Client가 Service와의 통신을 위해 필요한 Interface를 정의하는 일
- Service와 Client사이의 Interface는 IBinder 형태로 구현되어야 하며, IBinder 객체는 Service는 onBind() 콜백 메소드에서 Client에게 제공되어야 함
- Client가 IBinder 객체를 전달받으면, IBinder에 정의된 Interface를 통해 Service와 상호 동작이 가능

Bound Service 예제 0

- 다수의 Client가 한번에 Service와 결속(Binding)될 수 있음
- Client가 Service와 상호 동작할 때, 결속을 해제하려면 unBindService()를 호출하면 됨
- Service와 결속된 Client가 하나도 존재하지 않으면 시스템에서 Bound Service를 종료함

Bound Service 예제 0

- bindService() 메소드

```
public abstract boolean bindService
    (Intent service, ServiceConnection conn, int flags)
```

  - 첫 번째 파라미터는 service 대상을 가리킴
  - 두 번째 파라미터 conn은 서비스와 연결되거나 끊길 때의 콜백 인자
  - 세 번째 파라미터 flags는 Context의 상수 등을 넣어 서비스의 시작 방법 등을 정의
    - 가장 많이 쓰는 상수는 Context.BIND_AUTO_CREATE

Bound Service 예제 0

- bindService()에 사용되는 FLAG
  - BIND_AUTO_CREATE
    - Component와 연결되 있는 동안 비 정상적으로 종료 시 자동으로 다시 시작됨
  - BIND_DEBUG_UNBIND
    - 비 정상적으로 연결이 끊어지면 로그를 남김(디버깅용)
  - BIND_NOT_FOREGROUND
    - Background로만 동작
    - 만약 Activity에서 생성한 경우 Activity와 생성주기를 같이함

Bound 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"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="10dp"
        android:text="Bound Service 테스트"
        android:background="#FF00FF"
        android:textSize="20sp" />
```

Bound Service 예제 0

- 사용자 인터페이스

```
<EditText
    android:id="@+id/edit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="숫자 입력"
    android:inputType="number"/>
<Button
    android:id="@+id/start"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Bound Service 시작"/>

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

Bound Service 예제 0

- 사용자 인터페이스

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

Bound Service 예제 0

- MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    Button button2;
    BoundService boundService;
    boolean isService = false;
    static TextView textView;

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

        final Intent intent = new Intent(MainActivity.this, BoundService.class);
        final EditText editText = findViewById(R.id.edit);
        textView = findViewById(R.id.text);
```

Bound Service 예제 0

```
Button button1 = findViewById(R.id.start);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isService) {
            Toast.makeText(getBaseContext(), "Bound Service 중 입니다",
                                                Toast.LENGTH_SHORT).show();
        }else {
            String test = editText.getText().toString();
            if (test.equals("")) {
                Toast.makeText(getBaseContext(), "입력해주세요",
                                                 Toast.LENGTH_SHORT).show();
            } else {
                intent.putExtra("test", test);
                bindService(intent, connection, Context.BIND_AUTO_CREATE);
                button2.setEnabled(true);
                isService = true;
            }
        }
    }
});
```

Bound Service 예제 0

- MainActivity.JAVA

```
        button2 = findViewById(R.id.stop);
        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (isService) {
                    unbindService(connection);
                    button2.setEnabled(false);
                    isService = false;
                } else {
                    Toast.makeText(getBaseContext(),
                        "현재 서비스 중지 상태입니다", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
```

Bound Service 예제 0

- MainActivity.JAVA

```
    private ServiceConnection connection = new ServiceConnection() {
        @Override
        public void onServiceDisconnected(ComponentName name) {
            Toast.makeText(getBaseContext(), "서비스 해제",
                                                Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {
            BoundService.MyBinder myBinder = (BoundService.MyBinder) service;
            boundService = myBinder.getService();
            Toast.makeText(getBaseContext(), "서비스 시작",
                                                  Toast.LENGTH_SHORT).show();
        }
    };
}
```

Bound Service 예제 0

- BoundService.JAVA

```
public class BoundService extends Service {
    public class MyBinder extends Binder {
        public BoundService getService() {
            return BoundService.this;
        }
    }

    private IBinder mBinder = new MyBinder();

    @Override
    public void onCreate() {
        super.onCreate();
        Toast.makeText(this, "Bound Service onCreate() 메소드 호출",
                                                    Toast.LENGTH_SHORT).show();
    }
```

Bound Service 예제 0

- BoundService.JAVA

```
@Override
public IBinder onBind(Intent intent) {
    int value = Integer.parseInt(intent.getStringExtra("test"));
    MainActivity.textView.setText(String.format("%d * %d = %d", value,
                                                    value, value * value));
    return mBinder;
}

@Override
public void onRebind(Intent intent) {
    Toast.makeText(this,  "onRebind() 호출", Toast.LENGTH_SHORT).show();
    super.onRebind(intent);
}

@Override
public boolean onUnbind(Intent intent) {
    Toast.makeText(this, "서비스 해제", Toast.LENGTH_SHORT).show();
    return true;
}
```

Bound Service 예제 0

- BoundService.JAVA

```
    @Override
    public void onDestroy() {
        super.onDestroy();
        Toast.makeText(this, "onDestory() 호출", Toast.LENGTH_SHORT).show();
    }
}
```

Bound Service 예제 0

- Manifest.XML

```
 ..........................................................
        <service android:name=".BoundService" />

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

Bound Service 예제 0

- Binder 클래스 확장하기
  - 서비스를 오직 자신의 앱에서만 사용하고 여러 프로세스에서 작동하지 않는다면 Binder 클래스를 상속받아 구현

- Binder 클래스를 사용한 방법
  - 서비스안에서, 아래 중 하나를 수행하는 Binder 객체 생성
    - 다른 컴포넌트가 호출 할 public 메서드들을 포함
    - 다른 컴포넌트가 호출 할 public 메서드들을 포함하는 현재의 service 인스턴스를 리턴
    - 다른 컴포넌트가 호출 할 public 메서드들을 포함하는 서비스에 호스팅하는 다른 클래스의 인스턴스를 반환
  - onBind() 콜백 메소드에서 Binder 인스턴스를 리턴
  - 다른 컴포넌트는 onServiceConnected() 콜백 메소드로부터 이 Binder 객체를 받아 제공된 메소드들을 사용하여 서비스를 호출

Bound Service 예제 0

- MyBinder 클래스 생성
  - Binder 클래스를 상속받는 MyBinder 클래스를 생성해주고, 이 클래스 안에 클라이언트가 호출할 수 있는 public 타입의 getService() 메소드를 생성
  - 이 메소드가 return 해주는 값은 BoundService의 인스턴스

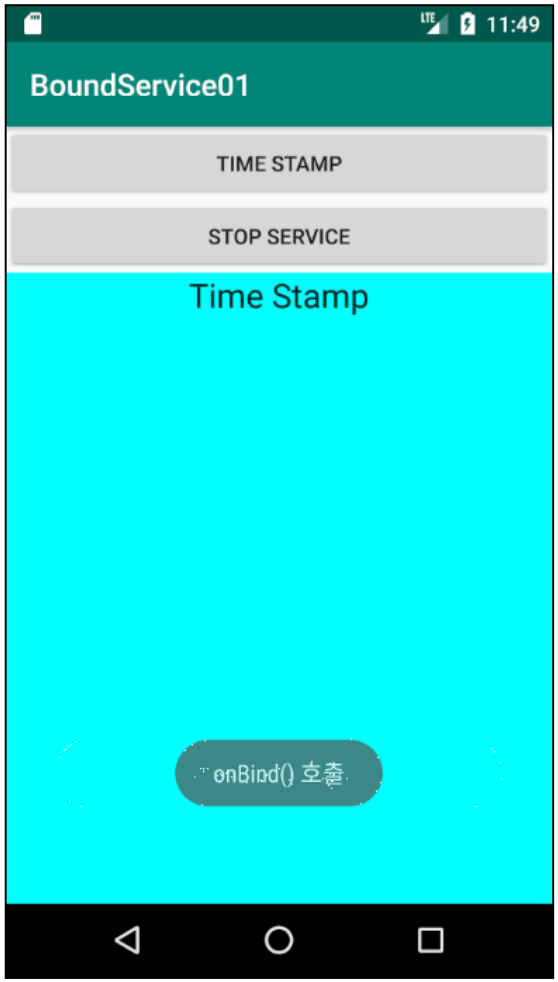
```
public class MyBinder extends Binder {
    public BoundService getService() {
        return BoundService.this;
    }
}
```

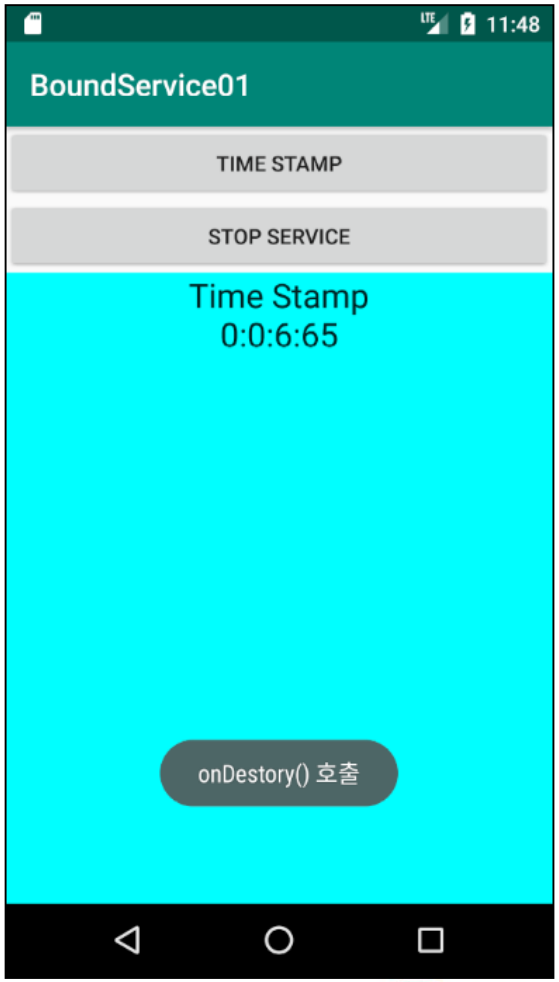
Bound Service 예제 0

- Service Binding 구현 & 동작
  - Service Binding 객체를 생성하려면 콜백 메소드인 onBind()를 구현해야 함
  - onBind()는 IBinder를 반환하는데, 바로 이 객체가 '서비스'와 '클라이언트' 사이의 인터페이스 역할
  - 클라이언트가 bindService()를 호출하면, 클라이언트가 서비스에 연결되면서 IBinder가 반환되고, 클라이언트가 IBinder를 받으면 이 인터페이스를 통해 주고 받는 것이 가능해지는 것
  - 서비스가 제공하는 다른 메소드 호출 가능
  - 여러 클라이언트가 하나의 서비스에 동시 접속이 가능하며, 클라이언트가 서비스와의 접속을 마치려면 unbindService()를 호출
  - 서비스에 연결된 클라이언트가 하나도 남아있지 않으면 서비스는 자동 소멸

Bound Service 예제 1

- Bound Service를 이용하여 Time Stamp를 만들어보자





Bound 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"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/print"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Time Stamp"/>
    <Button
        android:id="@+id/stop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Stop Service"/>
```

Bound Service 예제 1

- 사용자 인터페이스

```
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#00FFFF">

        <TextView
            android:id="@+id/timestamp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Time Stamp"
            android:gravity="center"
            android:textAppearance="?android:attr/textAppearanceLarge"/>
    </ScrollView>
</LinearLayout>
```

Bound Service 예제 1

- MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    Intent intent;
    BoundService mService;
    boolean serviceflag = false;

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

        intent = new Intent(this, BoundService.class);

        final TextView timestamp = findViewById(R.id.timestamp);
```

Bound Service 예제 1

- MainActivity.JAVA

```
Button button1 = findViewById(R.id.print);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!serviceflag) {
            bindService(intent, connection, Context.BIND_AUTO_CREATE);
            serviceflag = true;
            timestamp.setText("Time Stamp");
        } else {
            timestamp.setText(timestamp.getText() + "₩n" +
                                                mService.getTimestamp());
        }
    }
});
```

Bound Service 예제 1

- MainActivity.JAVA

```
    Button button2 = findViewById(R.id.stop);
    button2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (serviceflag) {
                unbindService(connection);
                serviceflag = false;
            }
        }
    });
}

@Override
protected void onStart() {
    super.onStart();
    bindService(intent, connection, Context.BIND_AUTO_CREATE);
    serviceflag = true;
}
```

Bound Service 예제 1

- MainActivity.JAVA

```
@Override
protected void onStop() {
    super.onStop();
    if (serviceflag) {
        unbindService(connection);
        serviceflag = false;
    }
}
```

Bound Service 예제 1

- MainActivity.JAVA

```
    private ServiceConnection connection = new ServiceConnection() {

        @Override
        public void onServiceDisconnected(ComponentName name) {
            serviceflag = false;
        }

        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {
            BoundService.MyBinder myBinder = (BoundService.MyBinder) service;
            mService = myBinder.getService();
            serviceflag = true;
        }
    };
}
```

Bound Service 예제 1

- BoundService.JAVA

```
public class BoundService extends Service {
    public class MyBinder extends Binder{
        public BoundService getService() {
            return BoundService.this;
        }
    }
    private IBinder mBinder = new MyBinder();
    private Chronometer chronometer;

    @Override
    public void onCreate() {
        super.onCreate();
        Toast.makeText(this, "Bound Service onCreate() 메소드 호출",
                                                    Toast.LENGTH_SHORT).show();
        chronometer = new Chronometer(this);
        chronometer.setBase(SystemClock.elapsedRealtime());
        chronometer.start();
    }
```

Bound Service 예제 1

- BoundService.JAVA

```
@Override
public IBinder onBind(Intent intent) {
    Toast.makeText(this, "onBind() 호출", Toast.LENGTH_SHORT).show();
    return mBinder;
}

@Override
public void onRebind(Intent intent) {
    Toast.makeText(this,  "onRebind() 호출", Toast.LENGTH_SHORT).show();
    super.onRebind(intent);
}

@Override
public boolean onUnbind(Intent intent) {
    Toast.makeText(this, "onUnbind() 호출", Toast.LENGTH_SHORT).show();
    return true;
}
```

Bound Service 예제 1

- BoundService.JAVA

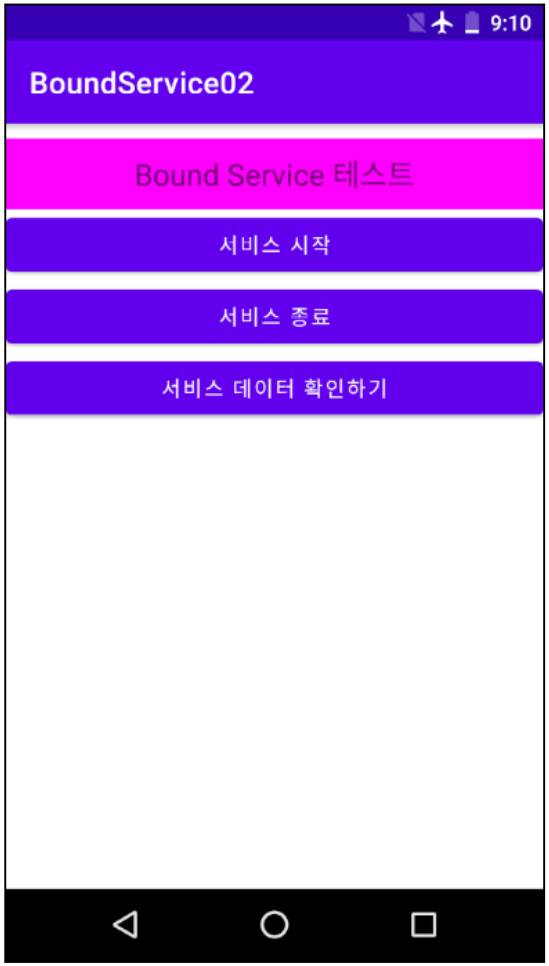
```
    @Override
    public void onDestroy() {
        super.onDestroy();
        Toast.makeText(this, "onDestory() 호출", Toast.LENGTH_SHORT).show();
        chronometer.stop();
    }
    public String getTimestamp() {
        long elapsedMillis = SystemClock.elapsedRealtime()
                - chronometer.getBase();
        int hours = (int) (elapsedMillis / 3600000);
        int minutes = (int) (elapsedMillis - hours * 3600000) / 60000;
        int seconds = (int) (elapsedMillis - hours * 3600000 -
                                            minutes * 60000) / 1000;
        int millis = (int) (elapsedMillis - hours * 3600000 - minutes * 60000
                                                - seconds * 1000);
        return hours + ":" + minutes + ":" + seconds + ":" + millis;
    }
}
```

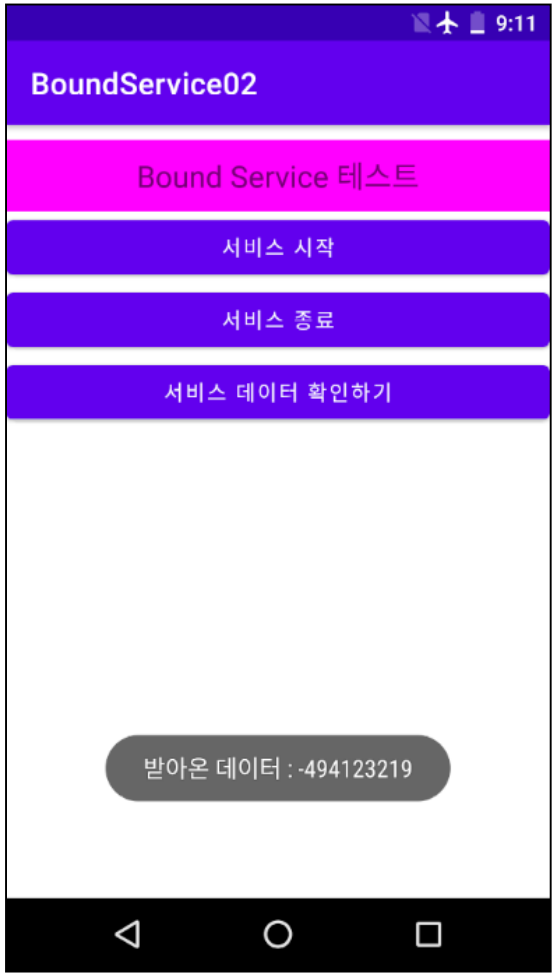
Bound Service 예제 1

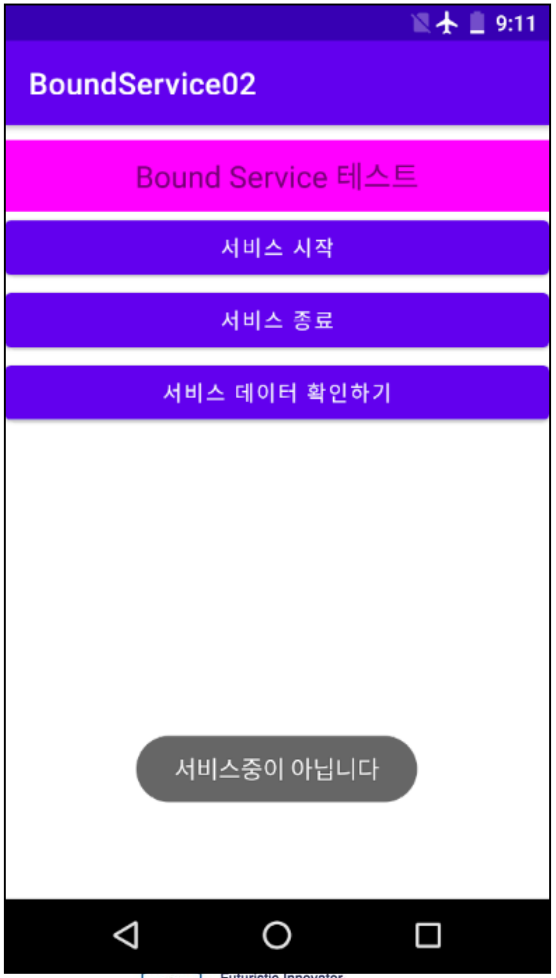
- Manifest.xml

```
        ..........................................................
        <activity android:name="com.example.boundservice01.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.boundservice01.BoundService"/>
    </application>

</manifest>
```

Bound Service 예제 2







Bound 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"
    tools:context=".MainActivity">

    <TextView
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="10dp"
        android:text="Bound Service 테스트"
        android:background="#FF00FF"
        android:textSize="20sp" />
```

Bound Service 예제 2

- 사용자 인터페이스

```
<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="서비스 시작" />

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="서비스 종료" />

<Button
    android:id="@+id/button3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="서비스 데이터 확인하기" />
</LinearLayout>
```

Bound Service 예제 2

- MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    MyService myService; // 서비스 객체
    boolean isService = false; // 서비스 중인 확인용

    ServiceConnection connection = new ServiceConnection() {
        public void onServiceConnected(ComponentName name, IBinder service) {
            MyService.MyBinder binder = (MyService.MyBinder) service;
            myService = binder.getService();
            isService = true;
        }

        public void onServiceDisconnected(ComponentName name) {
            isService = false;
        }
    };
```

Bound Service 예제 2

- MainActivity.JAVA

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

    Button button1 = findViewById(R.id.button1);
    button1.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, MyService.class);
            bindService(intent, connection, Context.BIND_AUTO_CREATE);
        }
    });
```

Bound Service 예제 2

- MainActivity.JAVA

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (!isService) {
            Toast.makeText(getApplicationContext(),
                    "서비스중이 아닙니다",
                    Toast.LENGTH_LONG).show();
        } else {
            unbindService(connection);
            isService = false;
        }
    }
});
```

Bound Service 예제 2

- MainActivity.JAVA

```
        Button button3 = findViewById(R.id.button3);
        button3.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {//서비스데이터확인
                if (!isService) {
                    Toast.makeText(getBaseContext(),
                            "서비스중이 아닙니다, 데이터받을수 없음",
                            Toast.LENGTH_LONG).show();
                }else {
                    Toast.makeText(getBaseContext(),
                            "받아온 데이터 : " + myService.getRan(),
                            Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

Bound Service 예제 2

- MyService.JAVA

```
public class MyService extends Service {
    IBinder mBinder = new MyBinder();

    class MyBinder extends Binder {
        MyService getService() { // 서비스 객체를 리턴
            return MyService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        // 액티비티에서 bindService() 를 실행하면 호출됨
        // 리턴한 IBinder 객체는 서비스와 클라이언트 사이의 인터페이스 정의한다
        return mBinder; // 서비스 객체를 리턴
    }
```

Bound Service 예제 2

- MyService.JAVA

```
    int getRan() { // 임의 랜덤값을 리턴하는 메서드
        return new Random().nextInt();
    }
    @Override
    public void onCreate() {
        super.onCreate();
    }
    public int onStartCommand(Intent intent, int flags, int startId) {
        return super.onStartCommand(intent, flags, startId);
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
    }
}
```

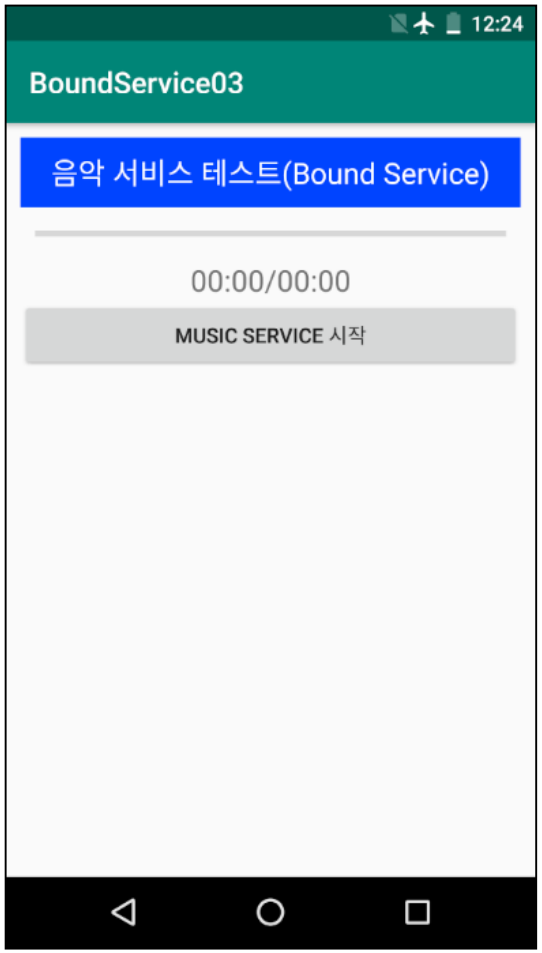
Bound Service 예제 2

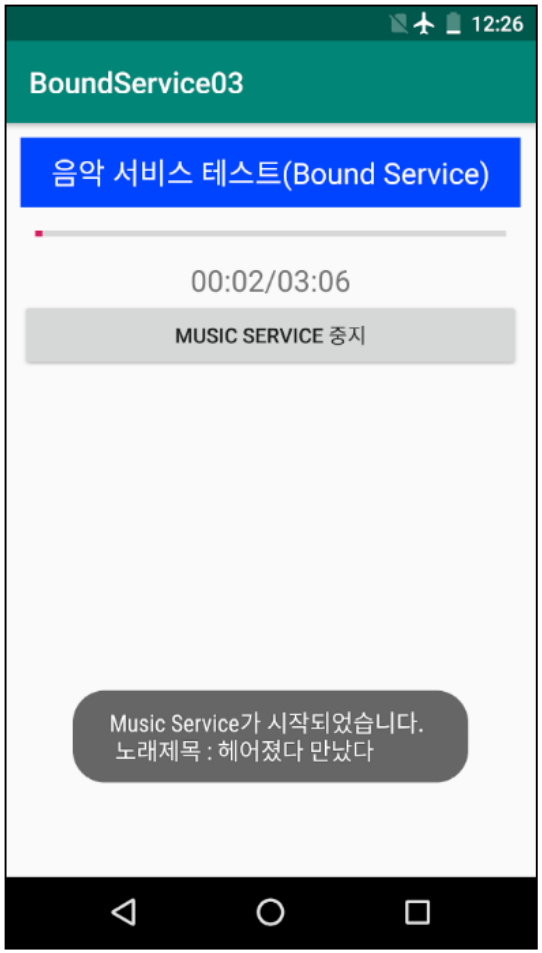
- 메니페스트

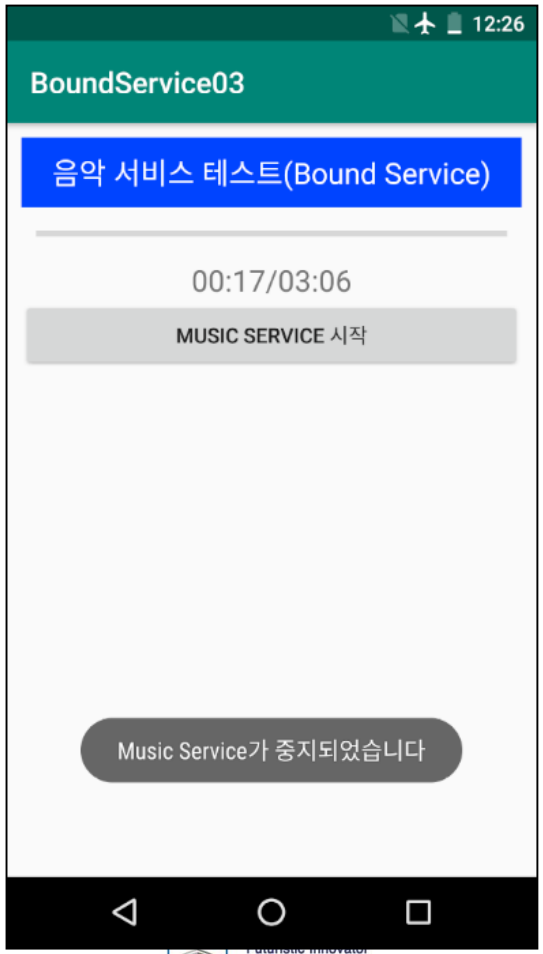
```
        <service
            android:name=".MyService"
            android:enabled="true"
            android:exported="false"></service>

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

Bound Service 예제 3(음악 연주)

- Bound Service를 이용하여 음악을 연주하는 프로그램을 만들어보자







Bound 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" />
```

Bound 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"
        android:textSize="20dp"/>
    <Button
        android:id="@+id/start"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Music Service 시작"/>
</LinearLayout>
```

Bound Service 예제 3(음악 연주)

- MyThread.JAVA

```
public class MyThread extends Thread {
    MediaPlayer player; // 음악 재생을 위한 객체
    int size;

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

Bound Service 예제 3(음악 연주)

```
    @Override
    public void run() {
        while (MainActivity.isService) {
            int current = player.getCurrentPosition();
            MainActivity.progressBar.setProgress(current);
            String msg = MusicService.getTime(current)+ "/" +
                                                    MusicService.getTime(size);
            try {
                Message message = MainActivity.handler.obtainMessage(0);
                Bundle bundle = new Bundle();
                bundle.putString("value", msg);
                message.setData(bundle);
                MainActivity.handler.sendMessage(message);
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                return;
            }
        }
    }
}
```

Bound Service 예제 3(음악 연주)

- MusicService.JAVA

```
public class MusicService extends Service {
    MediaPlayer player; // 음악 재생을 위한 객체
    int size;
    NotificationManager manager;
    private final int TEST_FLAG = 111;

   class MyBinder extends Binder {
        MusicService getService() {
            MusicService service = MusicService.this;
            return service;
        }
    }
```

Bound Service 예제 3(음악 연주)

- MusicService.JAVA

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
@Override
public IBinder onBind(Intent intent) {
    String title = intent.getStringExtra("title");
    Toast.makeText(this, "Music Service가 시작되었습니다.₩n 노래제목 : "
                                        +title, Toast.LENGTH_LONG).show();
    size = player.getDuration(); // 노래의 재생시간(miliSecond)
    player.start(); // 노래 시작
    manager.notify(TEST_FLAG, makeNotification());
    MainActivity.progressBar.setMax(size);
    MainActivity.textView.setText(getTime(player.getCurrentPosition())+
            "/" + getTime(size));
    MyThread thread = new MyThread(player, size);
    thread.start();
    return null;
}
```

Bound Service 예제 3(음악 연주)

- MusicService.JAVA

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

Bound Service 예제 3(음악 연주)

- MusicService.JAVA

```
@Override
public void onCreate() {
    super.onCreate();
    manager = (NotificationManager)getSystemService(
                                        Context.NOTIFICATION_SERVICE);
    player = MediaPlayer.create(this,R.raw.davichi);
    player.setLooping(true);
}

@Override
public void onRebind(Intent intent) {
    super.onRebind(intent);
}
```

Bound Service 예제 3(음악 연주)

- MusicService.JAVA

```
    @Override
    public void onDestroy() {
        super.onDestroy();
        manager.cancelAll();
        MainActivity.progressBar.setProgress(0);
        player.stop();  // 음악 종료
        player.release();
        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);
    }
}
```

Bound Service 예제 3(음악 연주)

- MainAcivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    static ProgressBar progressBar;
    static TextView textView;
    static boolean isService = false;

    ServiceConnection conn = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName componentName,
                                       IBinder iBinder) {
            MusicService.MyBinder binder = (MusicService.MyBinder) iBinder;
            isService = true;
        }
        @Override
        public void onServiceDisconnected(ComponentName componentName) {
            isService = false;
        }
    };
```

Bound Service 예제 3(음악 연주)

- MainAcivity.JAVA

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

    final Intent intent = new Intent(getApplicationContext(), MusicService.class);
    progressBar = findViewById(R.id.progress);
    textView = findViewById(R.id.status);

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

Bound Service 예제 3(음악 연주)

- MainAcivity.JAVA

```
                button.setText("Music Service 시작");
                unbindService(conn); // 서비스 종료
                isService = false;
            }
        }
    });
}
@SuppressLint("HandlerLeak")
static Handler handler = new Handler() {
    @Override
    public void handleMessage(@NonNull Message msg) {
        if (msg.what == 0) {
            Bundle bundle = msg.getData();
            String text = bundle.getString("value");
            textView.setText(text);
        }
    }
};
}
```

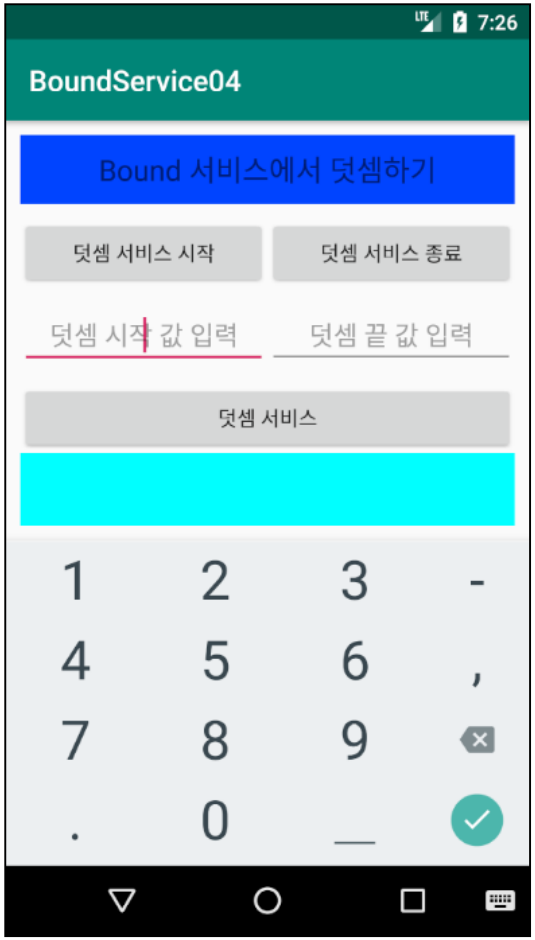
Bound Service 예제 3(음악 연주)

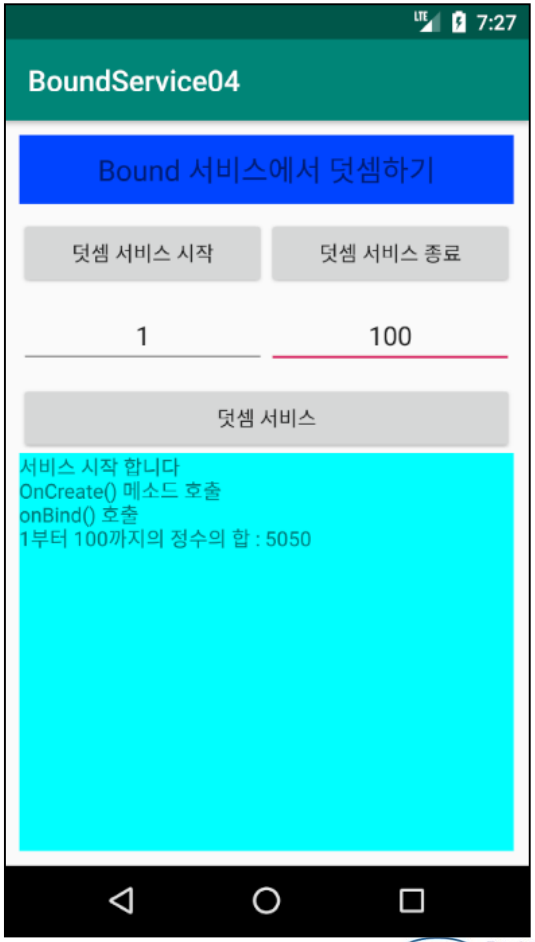
- 메니페스트

```
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name="com.example.boundservice03.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.boundservice03.MusicService"
            android:enabled="true"/>
    </application>

</manifest>
```

Bound Service 예제 4

- 두 수 from, to를 입력받아 from부터 to까지의 정수의 합을 구하는 것을 서비스를 이용하여 만들어보자.





Bound 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"
    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="Bound 서비스에서 덧셈하기"
        android:background="#0044FF"
        android:textSize="20sp" />
```

Bound Service 예제 4

- 사용자 인터페이스

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp">
    <Button
        android:id="@+id/start"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="덧셈 서비스 시작"/>
    <Button
        android:id="@+id/stop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="덧셈 서비스 종료"/>
</LinearLayout>
```

Bound Service 예제 4

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp">
    <EditText
        android:id="@+id/from"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center“
        android:inputType="number"
        android:layout_weight="1"
        android:hint="덧셈 시작 값 입력"/>
    <EditText
        android:id="@+id/to"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:inputType="number"
        android:layout_weight="1"
        android:hint="덧셈 끝 값 입력"/>
</LinearLayout>
```

Bound Service 예제 4

- 사용자 인터페이스

```
    <Button
        android:id="@+id/add"
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="덧셈 서비스"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:background="#00FFFF">
        <TextView
            android:id="@+id/result"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
    </ScrollView>
</LinearLayout>
```

Bound Service 예제 4

- MyService.JAVA

```
public class MyService extends Service {
    IBinder mBinder = new MyBinder();

    class MyBinder extends Binder {
        MyService getService() {
            MyService service;
            service = MyService.this;
            return service;
        }
    }

    @Override
    public void onCreate() {
        super.onCreate();
        MainActivity.result.append("OnCreate() 메소드 호출₩n");
    }
```

Bound Service 예제 4

- MyService.JAVA

```
    @Override
    public IBinder onBind(Intent intent) {
        MainActivity.result.append("onBind() 호출₩n");
        return mBinder;
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
        MainActivity.result.append("onDestory() 메소드₩n");
    }

    int addService(int from, int to) {
        int result = 0;
        for (int i = from; i <= to; i++)
            result += i;
        return result;
    }
}
```

Bound Service 예제 4

- MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    static TextView result;
    Button start, stop, add;
    EditText from, to;
    boolean isService = false;
    MyService service;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        result = findViewById(R.id.result);
        start = findViewById(R.id.start);
        stop = findViewById(R.id.stop);
        add = findViewById(R.id.add);
        from = findViewById(R.id.from);
        to = findViewById(R.id.to);
```

Bound Service 예제 4

- MainActivity.JAVA

```
start.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainActivity.this, MyService.class);
        if (!isService) {
            bindService(intent, conn, Context.BIND_AUTO_CREATE);
            result.append("서비스 시작 합니다₩n");
            Toast.makeText(getBaseContext(),
                    "서비스 시작 합니다", Toast.LENGTH_SHORT).show();
        } else {
            result.append("서비스 중 입니다₩n");
            Toast.makeText(getBaseContext(),
                    "서비스 중 입니다", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Bound Service 예제 4

- MainActivity.JAVA

```
stop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (!isService) {
            result.append("서비스중이 아닙니다₩n");
            Toast.makeText(getBaseContext(),
                    "서비스중이 아닙니다", Toast.LENGTH_SHORT).show();
            return;
        } else {
            result.append("서비스 종료 되었습니다₩n");
            Toast.makeText(getBaseContext(),
                "서비스 종료 되었습니다", Toast.LENGTH_SHORT).show();
            unbindService(conn);
            isService = false;
        }
    }
});
```

Bound Service 예제 4

- MainActivity.JAVA

```
add.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (!isService) {
            String msg = "서비스중이 아닙니다, 데이터 받을수 없음";
            result.append(msg + "₩n");
            Toast.makeText(getBaseContext(), msg,
                                        Toast.LENGTH_SHORT).show();
            return;
        }
        String one = from.getText().toString();
        String two = to.getText().toString();
        if (one.equals("") || two.equals("")) {
            String msg = "값을 입력해주세요";
            result.append(msg + "₩n");
            Toast.makeText(getBaseContext(), msg,
                                        Toast.LENGTH_SHORT).show();
```

Bound Service 예제 4

- MainActivity.JAVA

```
        } else {
            int first = Integer.parseInt(one);
            int second = Integer.parseInt(two);
            if (first > second) {
                String msg = String.format(Locale.KOREAN,
                  "%d가 %d보다 작으면 안됩니다", second, first);
                result.append(msg + "₩n");
                Toast.makeText(getBaseContext(), msg,
                                                Toast.LENGTH_SHORT).show();
            } else {
                int sum = service.addService(first, second);
                String msg = String.format(Locale.KOREAN,
                "%d부터 %d까지의 정수의 합 : %d", first, second, sum);
                result.append(msg + "₩n");
                Toast.makeText(getBaseContext(), msg,
                                                Toast.LENGTH_SHORT).show();
            }
        }
```

Bound Service 예제 4

- MainActivity.JAVA

```
            }
        });
    }

    ServiceConnection conn = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName componentName,
                                       IBinder iBinder) {
            MyService.MyBinder mb = (MyService.MyBinder) iBinder;
            service = mb.getService();
            isService = true;
        }
        @Override
        public void onServiceDisconnected(ComponentName componentName) {
            isService = false;
        }
    };
}
```

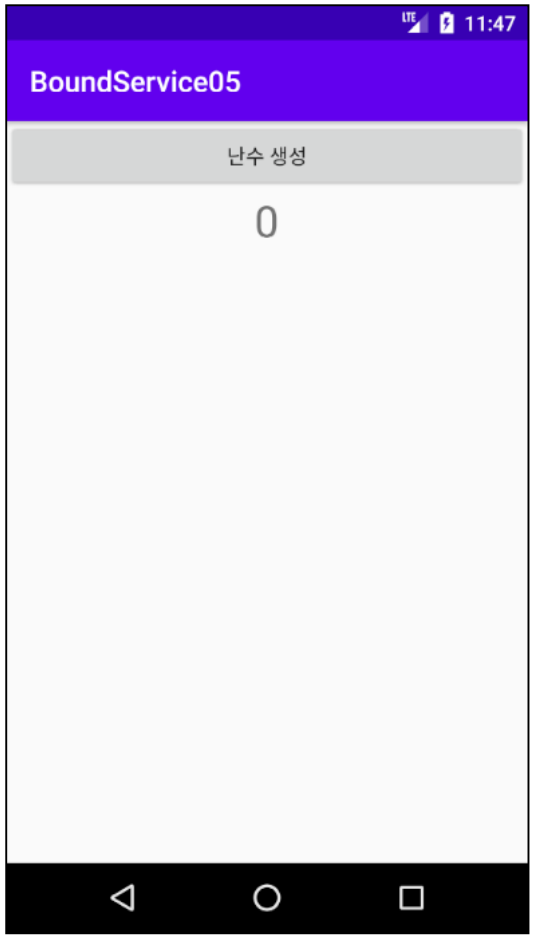
Bound Service 예제 4

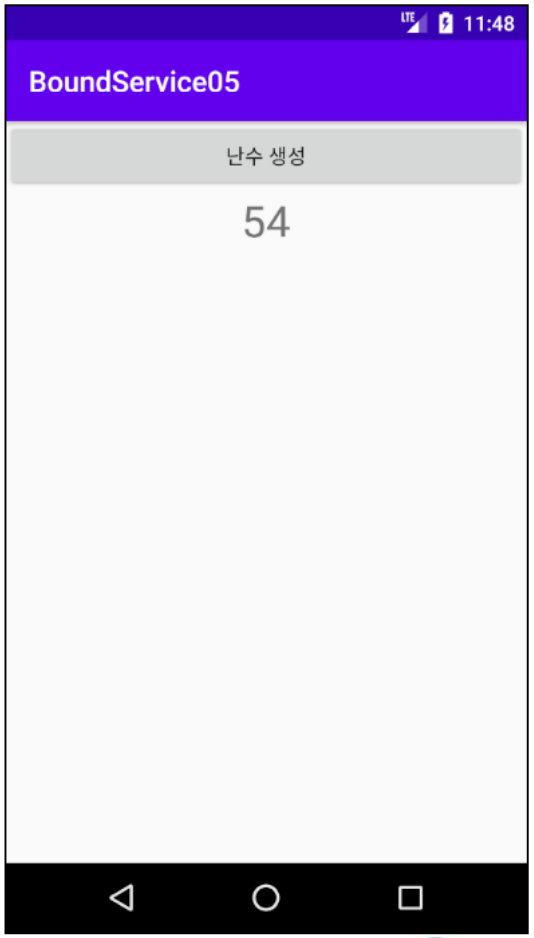
- 메니페스트 파일

```
        ....................................
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Bound Service 예제 5

- BoundService를 이용하여 난수를 발생하는 프로그램을 만들어보자





Bound 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onButtonClick"
        android:text="난수 생성" />
```

Bound Service 예제 5

- 사용자 인터페이스

```
    <TextView
        android:id="@+id/result"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:textSize="30dp"
        android:text="0"/>
</LinearLayout>
```

Bound Service 예제 5

- MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
    LocalService localService;
    boolean bound = false;
    TextView textView;

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

        textView = findViewById(R.id.result);
    }
    @Override
    protected void onStart() {
        super.onStart();
        Intent intent = new Intent(this, LocalService.class);
        bindService(intent, conect, Context.BIND_AUTO_CREATE);
    }
```

Bound Service 예제 5

- MainActivity.JAVA

```
private ServiceConnection conect = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName className,
                                   IBinder service) {
        LocalService.LocalBinder binder = (LocalService.LocalBinder) service;
        localService = binder.getService();
        bound = true;
    }
    @Override
    public void onServiceDisconnected(ComponentName arg0) {
        bound = false;
    }
};
```

Bound Service 예제 5

- MainActivity.JAVA

```
    @Override
    protected void onStop() {
        super.onStop();
        if (bound) {
            unbindService(conect);
            bound = false;
        }
    }

    public void onButtonClick(View v) {
        if (bound) {
            int num = localService.getRandomNumber();
            textView.setText(num + "");
        }
    }
}
```

Bound Service 예제 5

- LocalService.JAVA

```
public class LocalService extends Service {
    public LocalService() {
    }
    // 클라이언트에게 반환되는 바인더
    private final IBinder mBinder = new LocalBinder();
    // 난수 발생기
    private final Random mGenerator = new Random();

    // 클라이언트 바인더를 위한 클래스
    public class LocalBinder extends Binder {
        LocalService getService() {
            return LocalService.this;
        }
    }
```

Bound Service 예제 5

- LocalService.JAVA

```
    @Override
    public IBinder onBind(Intent intent) {
        return mBinder;
    }

    // 클라이언트를 위한 메소드
    public int getRandomNumber() {
        return mGenerator.nextInt(100);
    }
}
```

Bound Service 예제 5

- 메니페스트

```
        android:theme="@style/AppTheme">
        <service
            android:name="com.example.boundservice05.LocalService"
            android:enabled="true"
            android:exported="true"></service>

        <activity android:name="com.example.boundservice05.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```