



안드로이드 서비스

Android Service

4조

20060239 김영동
20060455 도홍렬
20060616 박효진

목차

1.서비스 클래스 사용하기

2.서비스 생애주기

3.서비스의 바인드

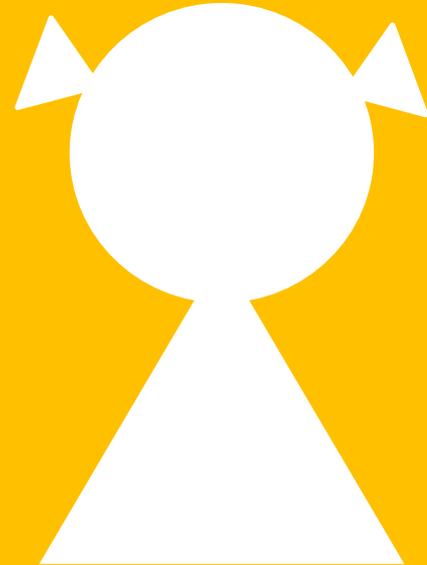
4.예 제



안드로이드 서비스란?

- 비주요한 사용자 인터페이스가 없음
- 액티비티와 연결되어 있지 않아도 끊임없이 실행이 됨
- 백그라운드에서 정해지지 않은 시간 동안 항상 동작
- 대표적인 예 : MP3 재생 어플리케이션

1.서비스 클래스 사용하기



```

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 void onStart(Intent intent, int startId){
        //서비스가 시작될때 수행할 동작
    }

    @Override
    public void onDestroy(Intent intent, int startId){
        //서비스가 종료될 때 수행할 동작
    }
}

```

- Service 클래스를 확장하여 새로운 클래스를 정의
- 해당 매소드를 오버라이딩

```
<service android:enable="true"  
    android:name=".MyService"> </service>
```

- 매니페스트 파일에 서비스를 선언
- application 노드에 자식 엘리먼트로 service를 추가

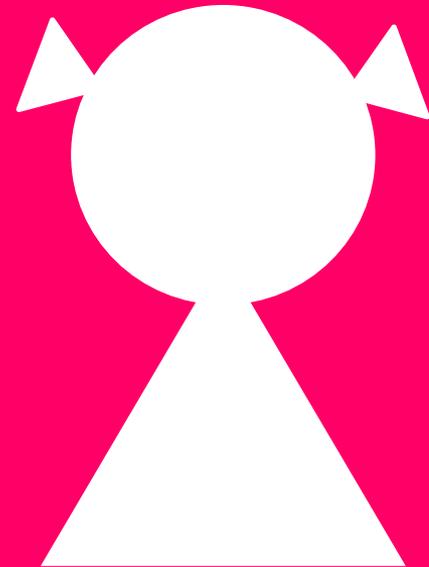
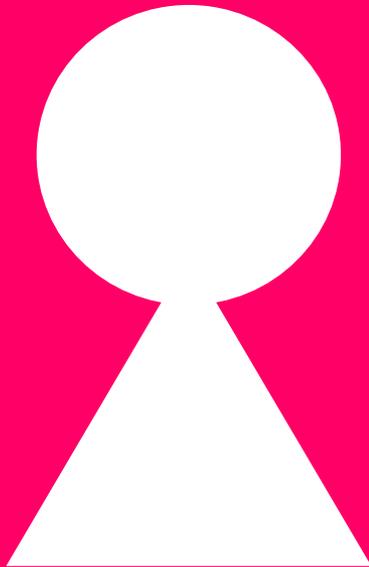
```
Intent intentMyService = new Intent(this, MyService.class)  
    ComponentName service = startService(intentMyService);
```

- 서비스를 시작하려면 서비스를 지정할 인텐트를 전달인자로 하여 startService() 메소드를 호출

```
stopService(new Intent(this, service.getClass()));
```

- 서비스를 정지시키려면 외부로부터 stopService() 메소드를 호출

2.서비스 생애주기



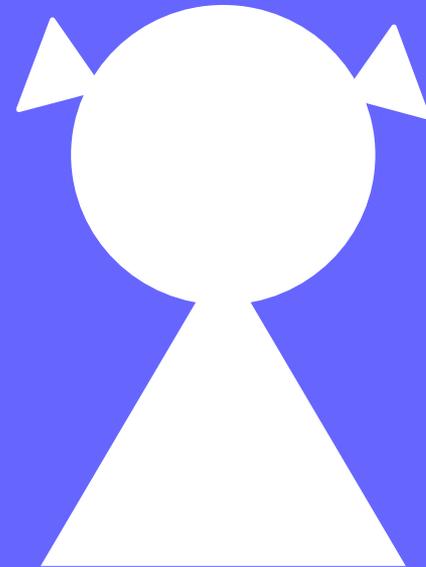
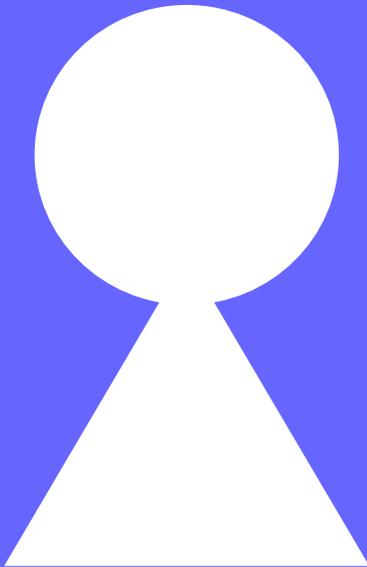
1. `void onCreate()`
2. `Void onStart(Intent intent)`
3. `Void onDesroy()`

- **서비스의 생애주기는 3개의 메소드로 진행됨**
- **onCreate() 메소드는 서비스가 생성 될 때 호출됨**
- **onStart() 메소드는 서비스를 외부에서 실행시키면 호출됨**
- **onDestroy() 메소드는 서비스가 종료될 때 호출됨**



- 이미 시작된 서비스는 onStart() 메소드의 실행을 마친 후 바로 onDestroy()로 가는 것이 아님
- 다음 실행을 위해 대기 상태로 들어가기 때문에 프로세스는 계속 살아있게 됨
- 이때 외부에서 서비스를 실행시키면 onStart() 메소드가 호출됨
- 시스템에 리소스가 모자라게 되면 OS가 프로세스를 죽이 되고 이때 onDestroy() 메소드에 의해서 서비스가 종료됨
- 서비스를 다시 시작하게 되면 onCreate()가 작동하게 됨

3.서비스의 바인드

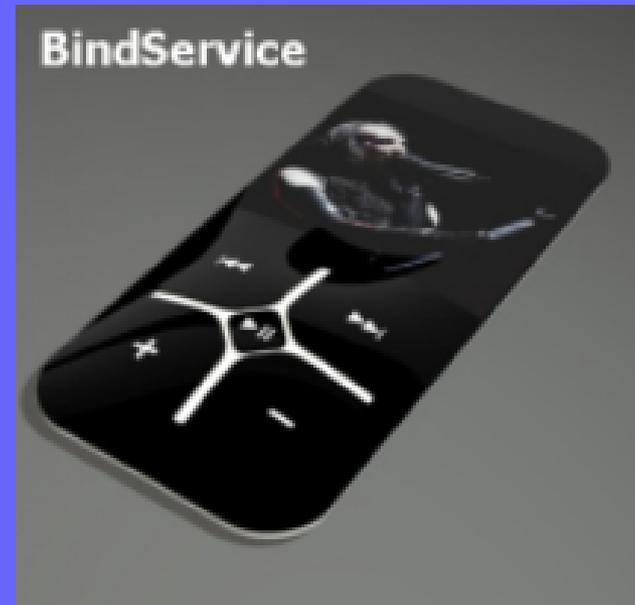
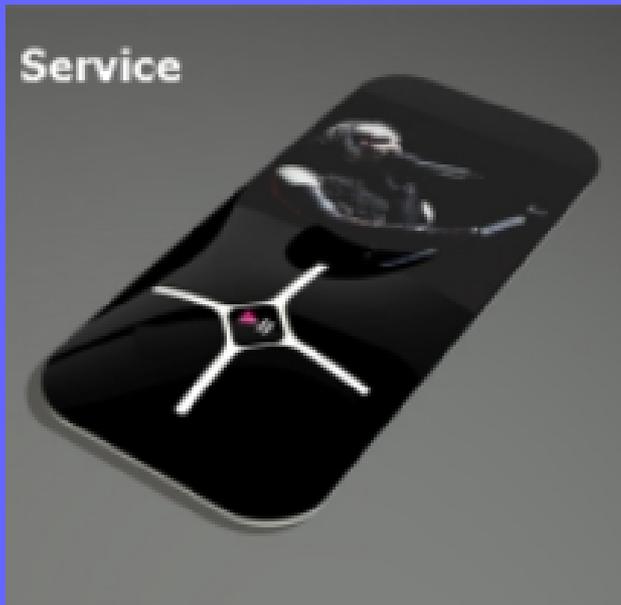


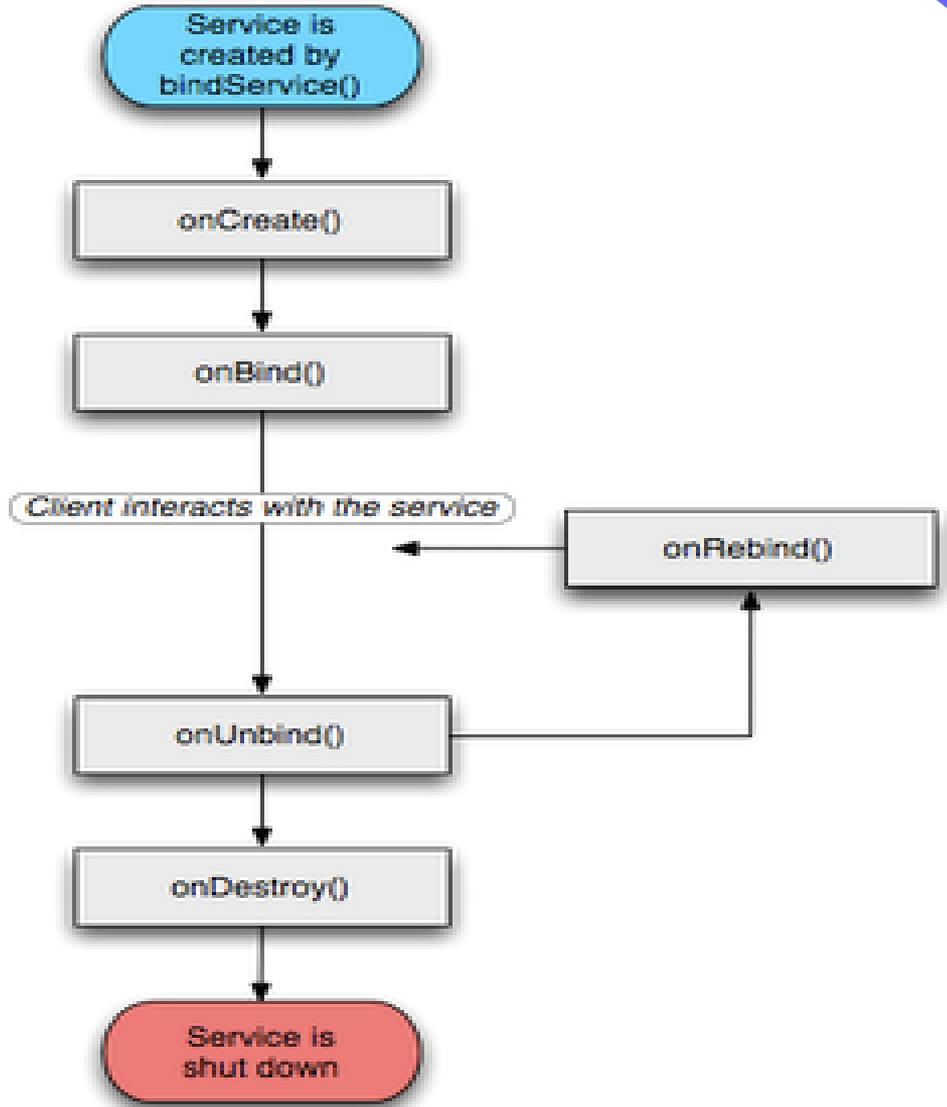
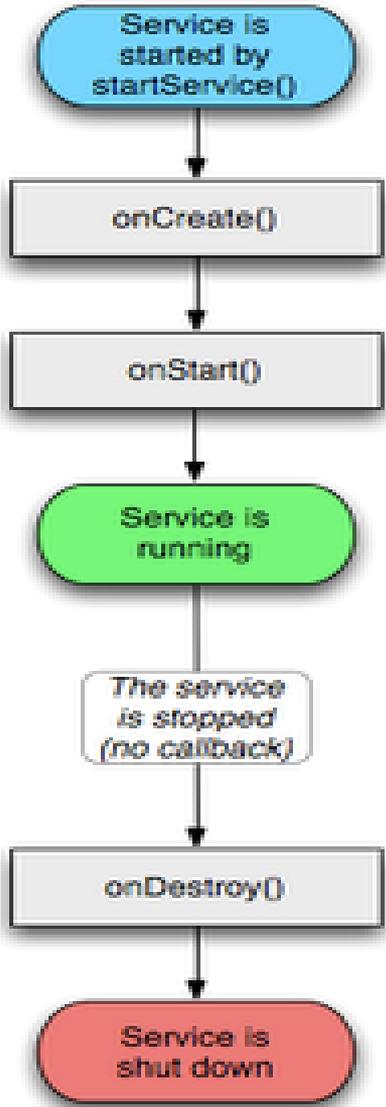
서비스의 바인드란?

- 서비스가 제공하는 기능을 호출하기 위하여 애플리케이션이 서비스를 접속 하는 것
- 서비스 기능을 호출함으로써 서비스로부터 데이터를 취득하거나 데이터를 설정할 수 있음

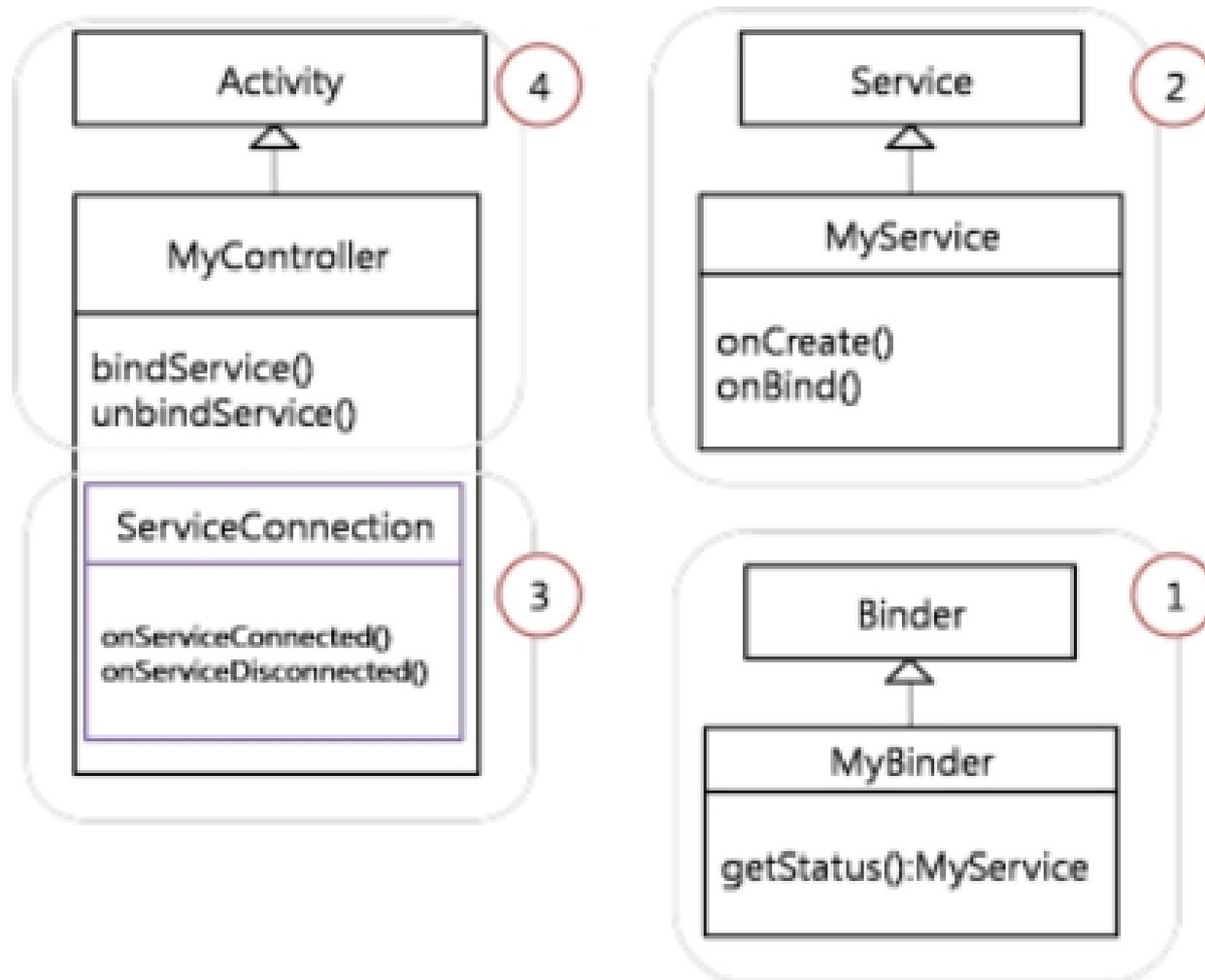
- 액티비티가 서비스에 바인딩되면, 그 액티비티는 해당 서비스 인스턴스 자체에 대한 레퍼런스를 유지함
- 인스턴스화된 기타 다른 클래스에 대해 하는 것처럼 실행 중인 서비스에 대해 메소드를 호출할 수 있도록 해줌
- 일반 서비스는 client가 서비스의 동작을 알 수 없지만, 바인딩된 서비스는 상태를 파악하고 제어할 수 있음

서비스 와 바인드 서비스

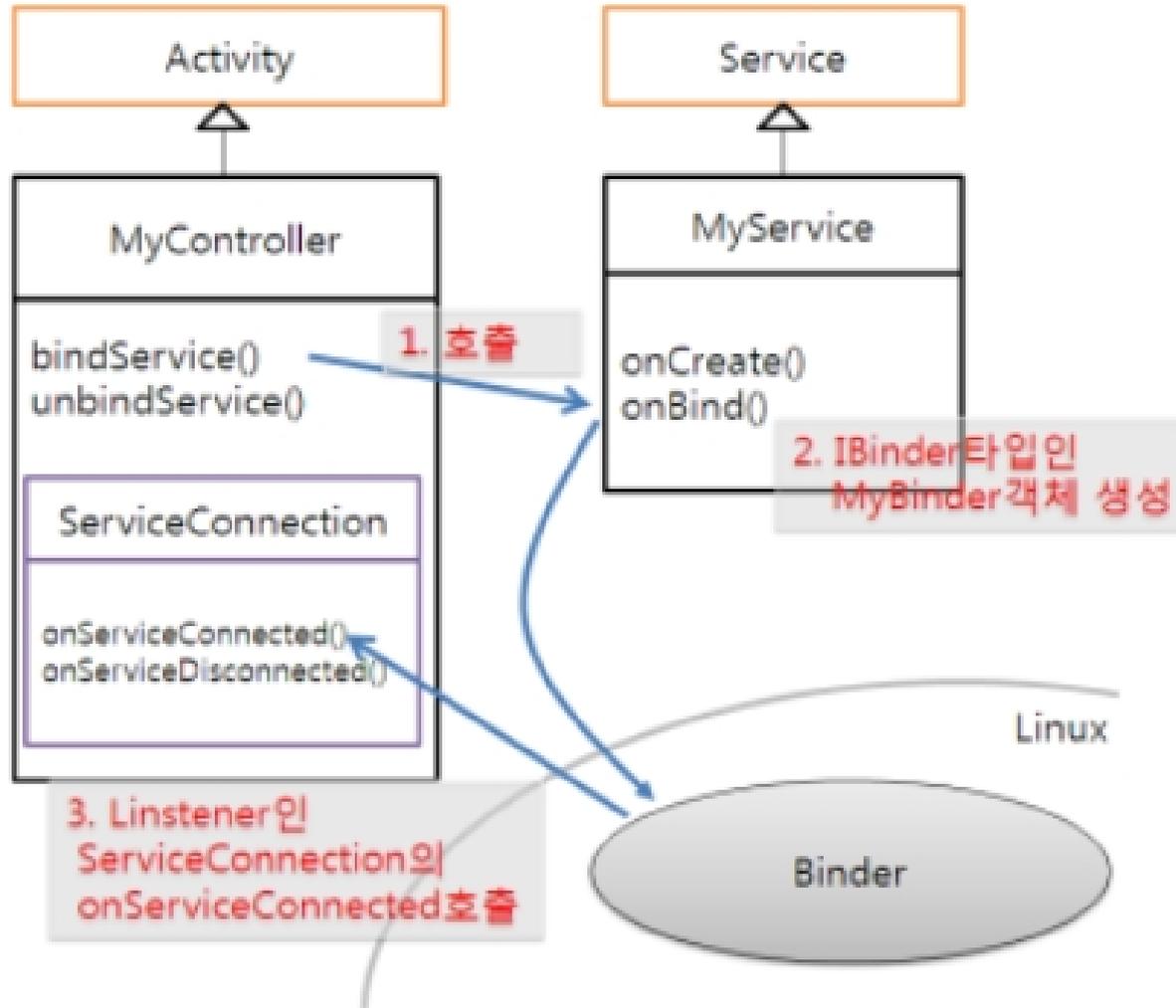




바인딩된 서비스 작성 순서



서비스 동작



```

public class MyService extends Service{

    private final IBinder binder = new MyBinder();

    @Override
    public IBinder onBind(Intent intent){
        Log.i(DEBUG_TAG, "MyService-onBind");
        // 서비스가 요구되었을 때에 Binder를 임플리먼트한 인스턴스를 반환한다.
        return binder;
    }

    public class MyBinder extends Binder{
        MyService getService(){
            Log.i(DEBUG_TAG, "MyBinder-getService");
            return MyService.this;
        }
    }
}

```

- 서비스의 바인딩을 위해서는 **onBind** 메소드 구현해야 함
- 서비스와 액티비티를 연결하여 서비스를 사용하는 클래스(액티비티)가 서비스 고유의 메소드를 호출하기 위해서 **Bind**의 요구를 실행함
- 바인드의 결과를 받는 **ServiceConnection**이라는 리스너를 지정함

```

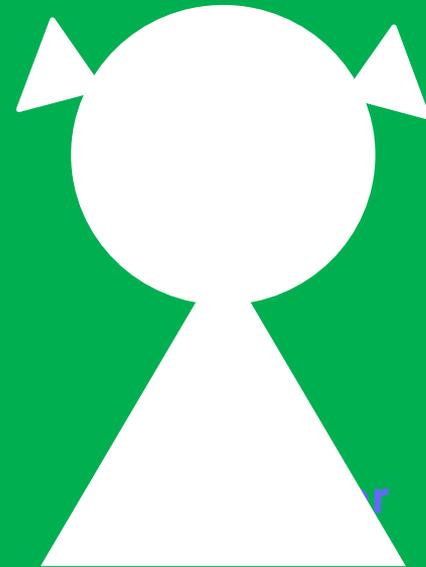
// 바인드할 서비스
private MyService serviceBider;

// 서비스와 액티비티 간의 연결을 다룬다
private ServiceConnection mConnection = new ServiceConnection(){
    //연결이 만들어질 때 호출된다
    @Override
    public void onServiceConnected(ComponentName name, IBinder service){
        serviceBinder = ((MyService.MyBinder)service).getService();
        Log.i(MyService.DEBUG_TAG, name.getShortClassName()+"Connected");
    }
    // 서비스가 예기치 않게 끊어진 경우 호출된다
    @Override
    public void onServiceDisconnected(ComponentName name){
        serviceBinder = null;
        Log.i(MyService.DEBUG_TAG ,name.getShortClassName()+"Disconnected");
    }
};

```

- 새로운 ServiceConnection을 구현하고 연결이 수립된 서비스 인스턴스에 대한 레퍼런스를 얻기 위해 OnServiceConnected()와 OnServiceDisconnected() 메소드를 오버라이딩해야 함
- 바인드가 성공하면 리스너의 onServiceConnected() 메소드가 호출됨
- 바인드하고 있는 불필요한 서비스를 상태에 따라 정지하기 위해서는 언바인딩함
- 그러면 onServiceDisconnected() 메소드가 호출 됨

4.예 제



EX.서비스 생애주기

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"/>
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/go"
    android:text="서비스 시작" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/stop"
    android:text="서비스 종료" />
</LinearLayout>
```

EX.서비스 생애주기

MyService.java ✖

```
package com.example.servicedemo;
```

```
import android.app.Service;  
import android.content.Intent;  
import android.os.IBinder;  
import android.util.Log;
```

```
public class MyService extends Service {  
    private static final String DEBUG_TAG = "ServiceDEBUG";
```

```
    @Override  
    public void onCreate() {  
        super.onCreate();  
        Log.i(DEBUG_TAG, "MyService-onCreat");  
    }
```

```
    @Override  
    public void onStart(Intent intent, int startId) {  
        Log.i(DEBUG_TAG, "MyService-onStart");  
    }
```

EX.서비스 생애주기

*MyService.java ✖

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

⊖    @Override
    public void onDestroy() {
        Log.i(DEBUG_TAG, "MyService-onDestroy");
        super.onDestroy();
    }
}
```

EX.서비스 생애주기

ServiceDemo.java ✕

```
package com.example.servicedemo;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class ServiceDemo extends Activity {
    ComponentName service;
    Intent intentMyService;
    TextView status;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

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

EX.서비스 생애주기

*ServiceDemo.java

```
public void onClick(View v) {
    intentMyService = new Intent(ServiceDemo.this, MyService.class);
    service = startService(intentMyService);
    Toast.makeText(ServiceDemo.this, "Starting...", Toast.LENGTH_SHORT).show();
}
});

Button stop = (Button) findViewById(R.id.stop);
stop.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Toast.makeText(ServiceDemo.this, "Stop...", Toast.LENGTH_SHORT).show();
        stopService(intentMyService);
    }
});
}
```

EX.서비스 생애주기

dem Manifest ☒

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.servicedemo"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".ServiceDemo"
            android:label="@string/app_name">
            <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=".MyService"></service>
    </application>
    <uses-sdk android:minSdkVersion="5" />

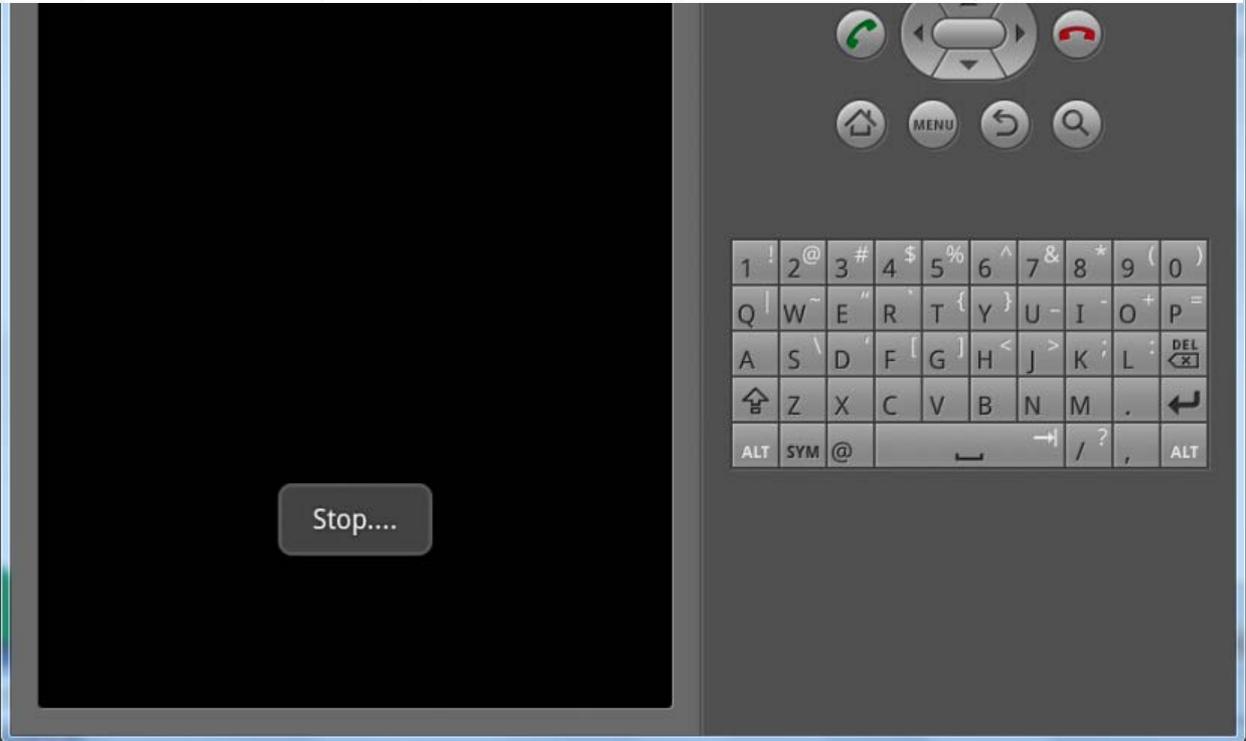
</manifest>
```

EX.서비스 생애주기

LogCat ☒

Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.

Level	Time	PID	Application	Tag	Text
D	04-11 13:03:55...	168	com.svox.pico	dalvikvm	GC freed 730 objects
I	04-11 13:03:58...	51	system_process	NotificationService	enqueueToast pkg=com.
I	04-11 13:03:58...	283	com.example.serv...	ServiceDEBUG	MyService-onCreat
I	04-11 13:03:58...	283	com.example.serv...	ServiceDEBUG	MyService-onStart
D	04-11 13:04:05...	106	android.process....	dalvikvm	GC freed 2360 objects
I	04-11 13:04:57...	51	system_process	NotificationService	enqueueToast pkg=com.
I	04-11 13:04:57...	283	com.example.serv...	ServiceDEBUG	MyService-onDestroy



EX.서비스에 액티비티 바인딩하기

MyService.java ✕

```
package com.example.servicedemo;
```

```
import android.app.Service;  
import android.content.Intent;  
import android.os.Binder;  
import android.os.IBinder;  
import android.util.Log;
```

```
public class MyService extends Service {  
    public static final String DEBUG_TAG = "ServiceDEBUG";
```

```
    @Override  
    public void onCreate() {  
        super.onCreate();  
        Log.i(DEBUG_TAG, "MyService-onCreat");  
    }
```

```
    @Override  
    public void onStart(Intent intent, int startId) {  
        Log.i(DEBUG_TAG, "MyService-onStart");  
    }
```

EX.서비스에 액티비티 바인딩하기

*MyService.java ☒

```
    @Override
    public IBinder onBind(Intent intent) {
        Log.i(DEBUG_TAG, "MyService-onBind");
        return binder;
    }

    @Override
    public void onDestroy() {
        Log.i(DEBUG_TAG, "MyService-onDestroy");
        super.onDestroy();
    }

    private final IBinder binder = new MyBinder();

    public class MyBinder extends Binder {
        MyService getService() {
            Log.i(DEBUG_TAG, "MyBinder-getService");
            return MyService.this;
        }
    }
}
```

EX.서비스에 액티비티 바인딩하기

ServiceDemo.java ✖

```
package com.example.servicedemo;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.IBinder;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class ServiceDemo extends Activity {
    Intent intentMyService;
    TextView status;
    // 바인드한 서비스
    private MyService serviceBinder;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

EX.서비스에 액티비티 바인딩하기

ServiceDemo.java ✕

```
@Override
protected void onStart() {
    super.onStart();
    // 서비스를 바인드한다.
    Intent bindIntent = new Intent(ServiceDemo.this, MyService.class);
    bindService(bindIntent, mConnection, Context.BIND_AUTO_CREATE);
    Log.i(MyService.DEBUG_TAG, "Starting...");
}

@Override
protected void onStop() {
    super.onStop();
    //바인드 종료하기
    this.unbindService(mConnection);
    Log.i(MyService.DEBUG_TAG, "Stop...");
}
// 서비스와 액티비티 간의 연결을 다룬다.
private ServiceConnection mConnection = new ServiceConnection() {
    // 연결이 만들어질 때 호출된다
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        serviceBinder = ((MyService.MyBinder)service).getService();
        Log.i(MyService.DEBUG_TAG, name.getShortClassName()+"Connected");
    }
}
```

EX.서비스에 액티비티 바인딩하기

*ServiceDemo.java ✕

```
// 서비스와 액티비티 간의 연결을 다룬다.
```

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

```
    // 연결이 만들어질 때 호출된다
```

```
    @Override
```

```
    public void onServiceConnected(ComponentName name, IBinder service) {
```

```
        serviceBinder = ((MyService.MyBinder)service).getService();
```

```
        Log.i(MyService.DEBUG_TAG, name.getShortClassName()+"Connected");
```

```
    }
```

```
    // 서비스가 예기치 않게 끊어진 경우 호출된다.
```

```
    @Override
```

```
    public void onServiceDisconnected(ComponentName name) {
```

```
        serviceBinder = null;
```

```
        Log.i(MyService.DEBUG_TAG, name.getShortClassName()+"Disconnected");
```

```
    }
```

```
};
```

```
}
```

EX.서비스에 액티비티 바인딩하기

The screenshot displays an Android emulator window titled "5554:7777". The main screen shows a service interface with the text "Service, Demo01" and "Hello World, ServiceDemo!". Below the text is a virtual keypad with various icons including camera, volume, power, call, navigation, home, menu, back, and search.

Below the emulator is the LogCat window, which shows a search bar and a table of log messages. The search bar contains the text "Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope." and a "ve" button.

Level	Time	PID	Application	Tag	Text
I	04-11 13:22:51...	204	com.example.serv...	ServiceDEBUG	Starting....
I	04-11 13:22:51...	204	com.example.serv...	ServiceDEBUG	MyService-onCreat
I	04-11 13:22:51...	204	com.example.serv...	ServiceDEBUG	MyService-onBind
I	04-11 13:22:51...	204	com.example.serv...	ServiceDEBUG	MyBinder-getService
I	04-11 13:22:51...	204	com.example.serv...	ServiceDEBUG	.MyServiceConnected
W	04-11 13:33:17...	204	com.example.serv...	KeyCharacterMap	No keyboard for id 0
W	04-11 13:33:17...	204	com.example.serv...	KeyCharacterMap	Using default keymap: /s
I	04-11 13:33:20...	204	com.example.serv...	ServiceDEBUG	Stop....
I	04-11 13:33:20...	204	com.example.serv...	ServiceDEBUG	MyService-onDestroy



THANK YOU FOR LISTENING

Q&A