CHAP 10. 서비스와 방송 수신자

서비스

- 사용자 인터페이스 없이 백그라운드에서 실행되는 컴포넌트
  - 배경 음악을 재생
  - 웹 사이트에서 주기적으로 데이터를 읽는다.
  - 주기적으로 폰의 사용량을 계산
  - 애플리케이션의 업데이트를 주기적으로 검사

실제 실행중인 서비스


SKT
3G
77%
오후 2:05
애플리케..
사용 중인 서비스 표시
SD 카드
실행 중
전체
환경설정
1개의 프로세스 및 0개의 서비스
49 MB
네이버백신
0개의 프로세스 및 1개의 서비스
0.00 B
다시 시작하는 중...
위메프
1개의 프로세스 및 1개의 서비스
1.7 MB
48:11:23
카카오톡
1개의 프로세스 및 2개의 서비스
15 MB
48:11:49
도돌 폰
1개의 프로세스 및 1개의 서비스
7.9 MB
48:11:21
네이버
1개의 프로세스 및 1개의 서비스
6.1 MB
47:45:44
AndroZip File Manager
1개의 프로세스 및 1개의 서비스
1.4 MB
48:11:21
RAM
741 MB 사용
91 MB 사용 가능

서비스의 종류

- 시작 타입의 서비스**(started service)**
  - 액티비티가 startService()를 호출하여서 서비스를 시작
- 연결 타입의 서비스**(bound service)**
  - 액티비티가 bindService()를 호출하여서 서비스를 시작

서비스 예제

- 배경에서 음악을 연주하는 서비스를 작성하여 보자.

서비스 사용 예제

- 앞의 서비스를 사용하는 예제를 작성
- 먼저 다음과 같은 인터페이스를 XML로 작성

리소스 준비

- mp3 형식의 음악 파일을 하나 다운로드받아서 /res/raw 디렉토리에 old_pop.mp3와 같은 이름으로 저장한다.

사용자 인터페이스

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="20dp"
        android:text="음악 서비스 테스트"
        android:textSize="20sp" />

    <Button
        android:id="@+id/start"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="시작" >
    </Button>

    <Button
        android:id="@+id/stop"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="중지" >
    </Button>

</LinearLayout>
```

MusicserviceTest.java

```
...
public class MainActivity extends AppCompatActivity implements OnClickListener  {
  private static final String TAG = "MusicServiceTest";
  Button start, stop;

  @Override
  public void onCreate(Bundle savedInstanceState)  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    start = (Button) findViewById(R.id.start);
    stop = (Button) findViewById(R.id.stop);
    start.setOnClickListener(this);
    stop.setOnClickListener(this);
}
```

MusicserviceTest.java

```
public void onClick(View src) {
  switch (src.getId())  {
  case R.id.start:
    Log.d(TAG, "onClick() start ");
    startService(new Intent(this, MusicService.class));
    break;

  case R.id.stop:
    Log.d(TAG, "onClick() stop");
    stopService(new Intent(this, MusicService.class));
    break;
  }
 }
}
```

음악을 연주하는 서비스

```
public class MusicService extends Service {

        private static final String TAG = "MusicService";
        MediaPlayer player;

        public IBinder onBind(Intent intent) {
                return null;
        }
        public void onCreate() {
                Log.d(TAG, "onCreate()");
                player = MediaPlayer.create(this, R.raw.old_pop);
                player.setLooping(false); // Set looping
        }
```

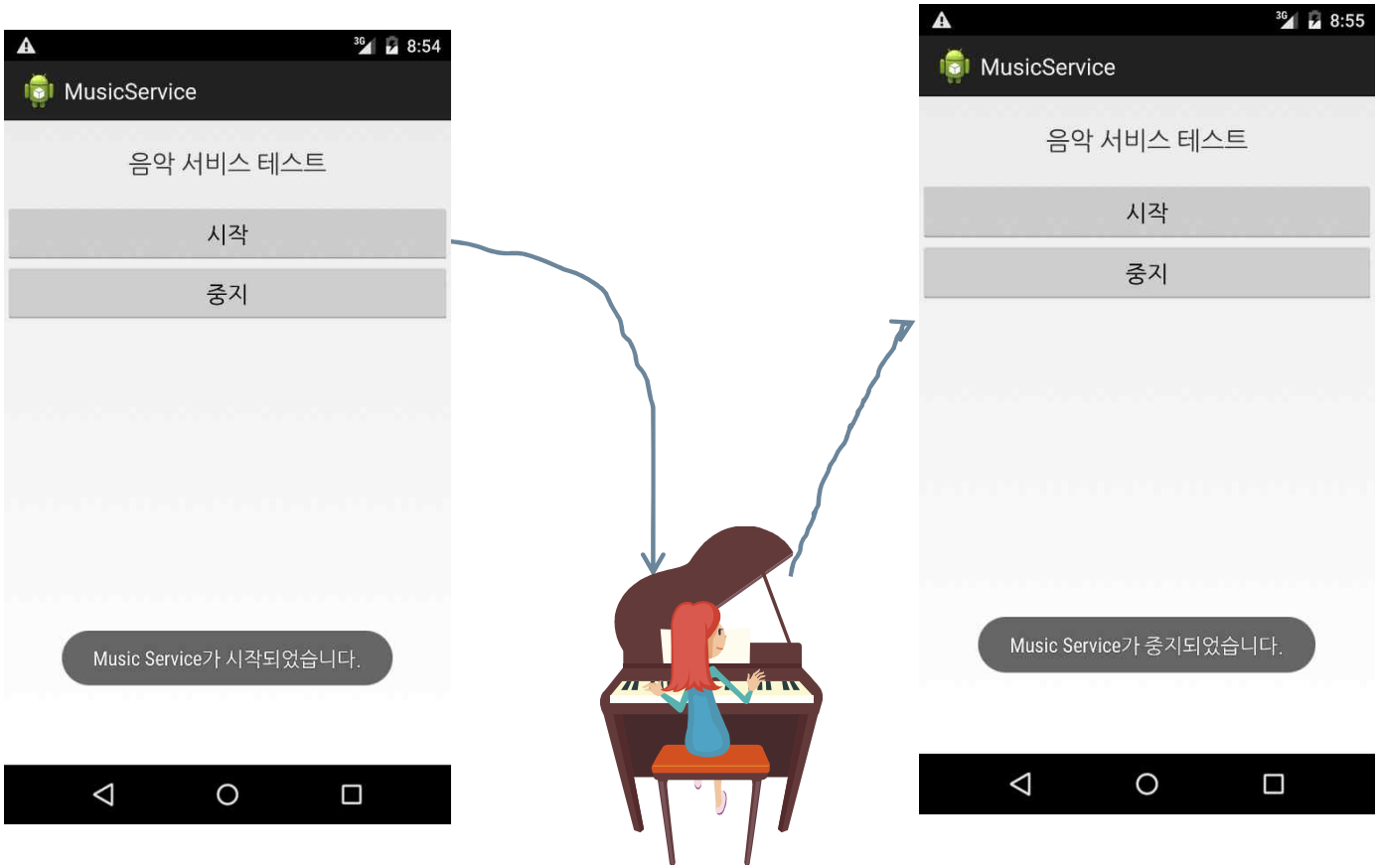
음악을 연주하는 서비스

```
        public void onDestroy() {
                Toast.makeText(this, "Music Service가 중지되었습니다.",
                        Toast.LENGTH_LONG).show();
                Log.d(TAG, "onDestroy()");
                player.stop();
        }
        public int onStartCommand(Intent intent, int flags, int startId) {
                Toast.makeText(this, "Music Service가 시작되었습니다.",
                        Toast.LENGTH_LONG).show();
                Log.d(TAG, "onStart()");
                player.start();
                return super.onStartCommand(intent, flags, startId);
        }
}
```

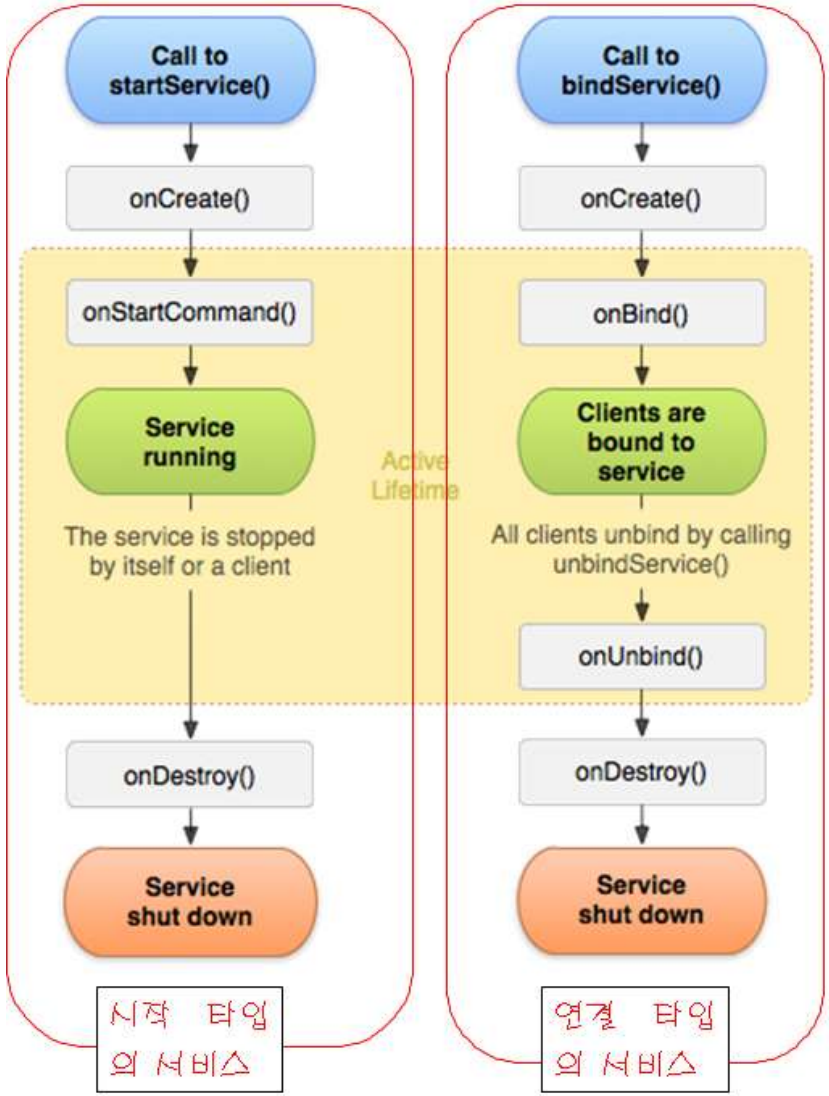
매니페스트 파일

```
<manifest
        ...
<application
        <activity          ...          </activity>
        <service
                android:enabled="true"
                android:name=".MusicService"
        />
</application>
</manifest>
```

실행 결과


8:54
MusicService
음악 서비스 테스트
시작
중지
Music Service가 시작되었습니다.
8:55
MusicService
음악 서비스 테스트
시작
중지
Music Service가 중지되었습니다.

서비스의 생애주기


Call to startService()
onCreate()
onStartCommand()
Service running
The service is stopped by itself or a client
onDestroy()
Service shut down
시작 타입의 서비스
Call to bindService()
onCreate()
onBind()
Clients are bound to service
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
Service shut down
연결 타입의 서비스
Active Lifetime

IntentService 클래스

- **Service 클래스 :** 만약 서비스에서 하는 작업이 상당히 시간을 많이 요구하는 작업이라면 서비스 안에서 새로운 스레드를 생성하는 것이 좋다.

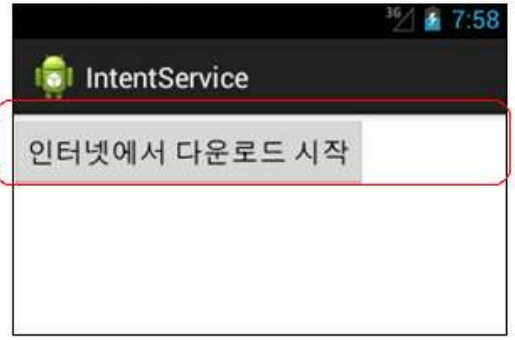
- **IntentService 클래스:**이 클래스는 시작 요청이 들어올 때마다 이것을 처리하는 작업 스레드를 별도로 생성한다.

예제: 인터넷으로부터 다운로드

사용자 인터페이스

main.xml

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

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="인터넷에서 다운로드 시작" />

</LinearLayout>
```



```
package kr.co.company.intentservice;
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

public class MainActivity extends AppCompatActivity {
    private Handler handler = new Handler() {
        public void handleMessage(Message message) {
            Object path = message.obj;
            if (message.arg1 == RESULT_OK && path != null) {
                Toast.makeText(getApplicationContext(),
                        "   "   +   path.toString()   +   "을   다운로드하였음.",
Toast.LENGTH_LONG)
                        .show();
            } else {
                Toast.makeText(getApplicationContext(), "다운로드 실패",
                        Toast.LENGTH_LONG).show();
            }
        };
    };
```

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

    public void onClick(View view) {
        Intent intent = new Intent(this, DownloadService.class);
        Messenger messenger = new Messenger(handler);
        intent.putExtra("MESSENGER", messenger);
        intent.setData(Uri.parse("https://www.naver.com/"));
        intent.putExtra("urlpath", "https://www.naver.com/");
        startService(intent);
    }
}
```

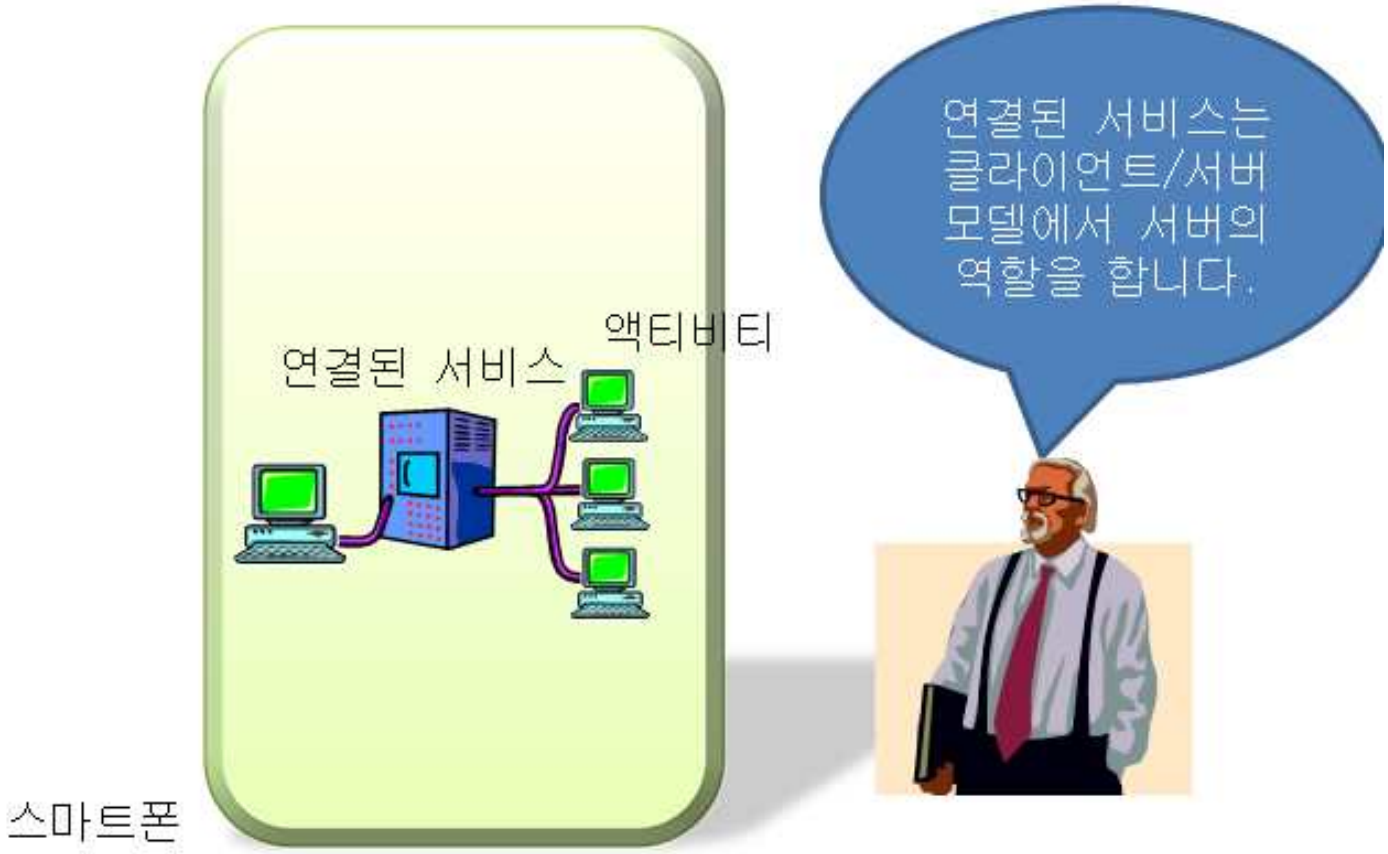
```
public class DownloadService extends IntentService {

   private int result = Activity.RESULT_CANCELED;

   public DownloadService() {
      super("DownloadService");
   }

   @Override
   protected void onHandleIntent(Intent intent) {
      Uri data = intent.getData();
      String urlPath = intent.getStringExtra("urlpath");
      String buffer=" ";

      InputStream stream = null;
      try {

         URL url = new URL(urlPath);
         stream = url.openConnection().getInputStream();
         InputStreamReader reader = new InputStreamReader(stream);
         int i = 0, next = -1;
         while ((next = reader.read()) != -1) {
            buffer += " " + (char)next;
            if( ++i > 100 ) break;
         }
         result = Activity.RESULT_OK;

      } catch (Exception e) {
         e.printStackTrace();
      } finally {
         if (stream != null) {
```

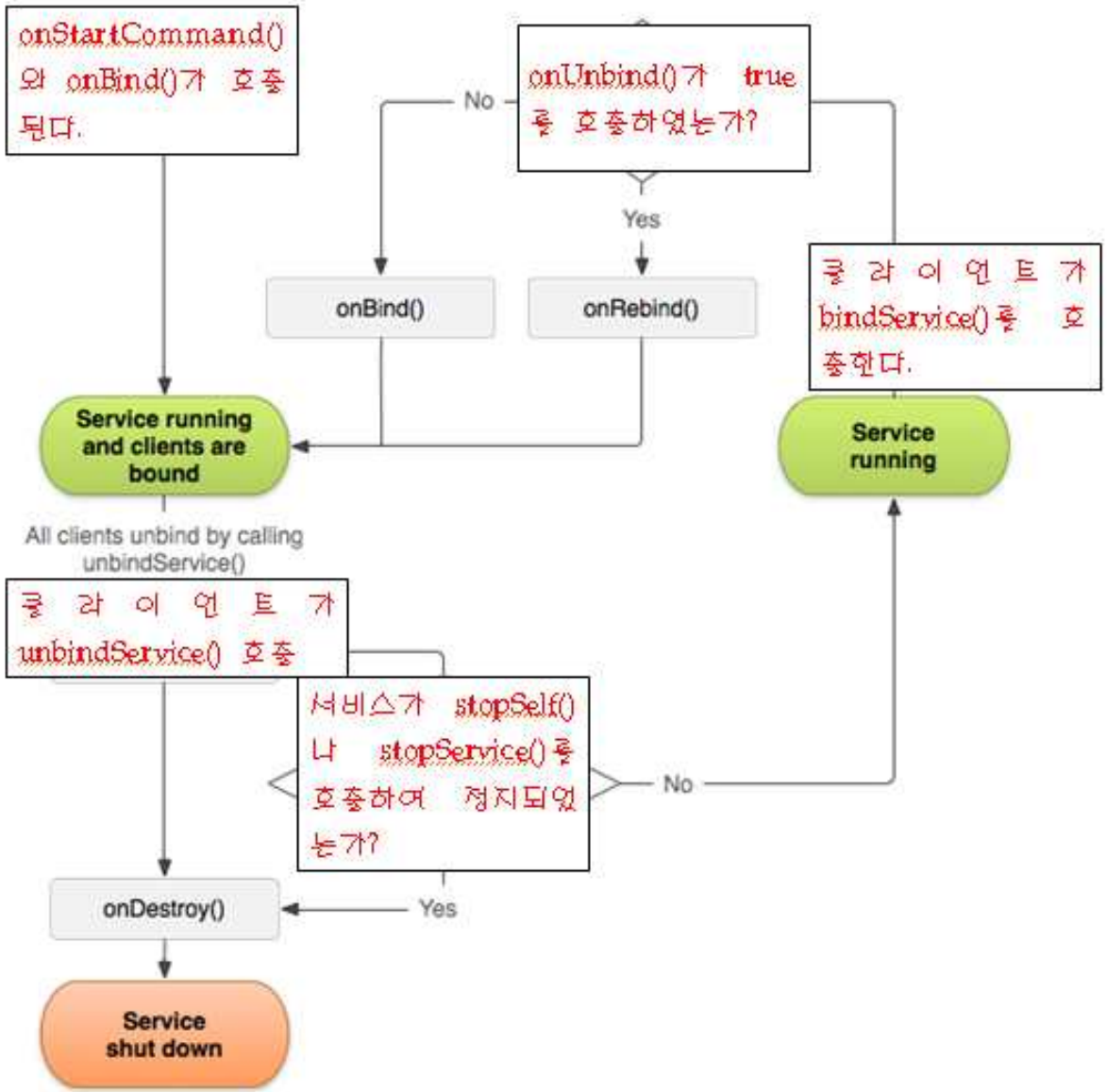
```
            try {
                stream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    Bundle extras = intent.getExtras();
    if (extras != null) {
        Messenger messenger = (Messenger) extras.get("MESSENGER");
        Message msg = Message.obtain();
        msg.arg1 = result;
        msg.obj = buffer;
        try {
            messenger.send(msg);
        } catch (android.os.RemoteException e1) {
            Log.w(getClass().getName(), "Exception sending message", e1);
        }

    }
  }
}
```

실행 결과


8:58
IntentService
인터넷에서 다운로드 시작
8:58
인터넷에서 다운로드 시작
<!doctype html><html item scope="" itemtype="http://schema.org/WebPage" lang="ko"><head><meta content=을 다운로드하였음.
->

연결 타입의 서비스



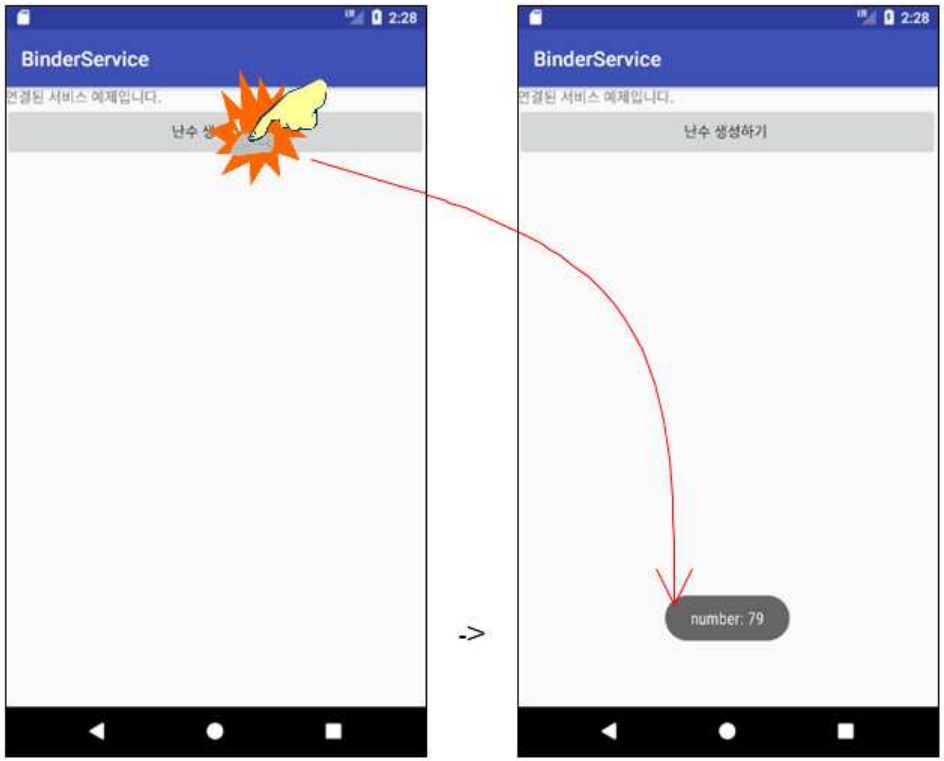

onStartCommand()와 onBind()가 호출된다.
onUnbind()가 true를 호출하였는가?
No
Yes
onBind()
onRebind()
클라이언트가 bindService()를 호출한다.
Service running and clients are bound
Service running
All clients unbind by calling unbindService()
클라이언트가 unbindService() 호출
서비스가 stopSelf()나 stopService()를 호출하여 정지되었는가?
No
Yes
onDestroy()
Service shut down

서비스 인터페이스를 정의하는 3가지의 방법

- **Binder** 클래스 확장하기
- **Messenger** 클래스 사용하기
- **Using AIDL**

Binder 클래스 확장하기 예제

- 난수를 발생시키서 다른 컴포넌트에게 서비스하는 연결된 타입의 서비스를 작성하여 보자.



사용자 인터페이스

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="연결된 서비스 예제입니다. " />

    <Button
        android:id="@+id/Button01"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onButtonClick"
        android:text="난수 생성하기 " >
    </Button>

</LinearLayout>
```

난수 발생 서비스 작성

```
public class LocalService extends Service {

// 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;
        }
    }

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

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

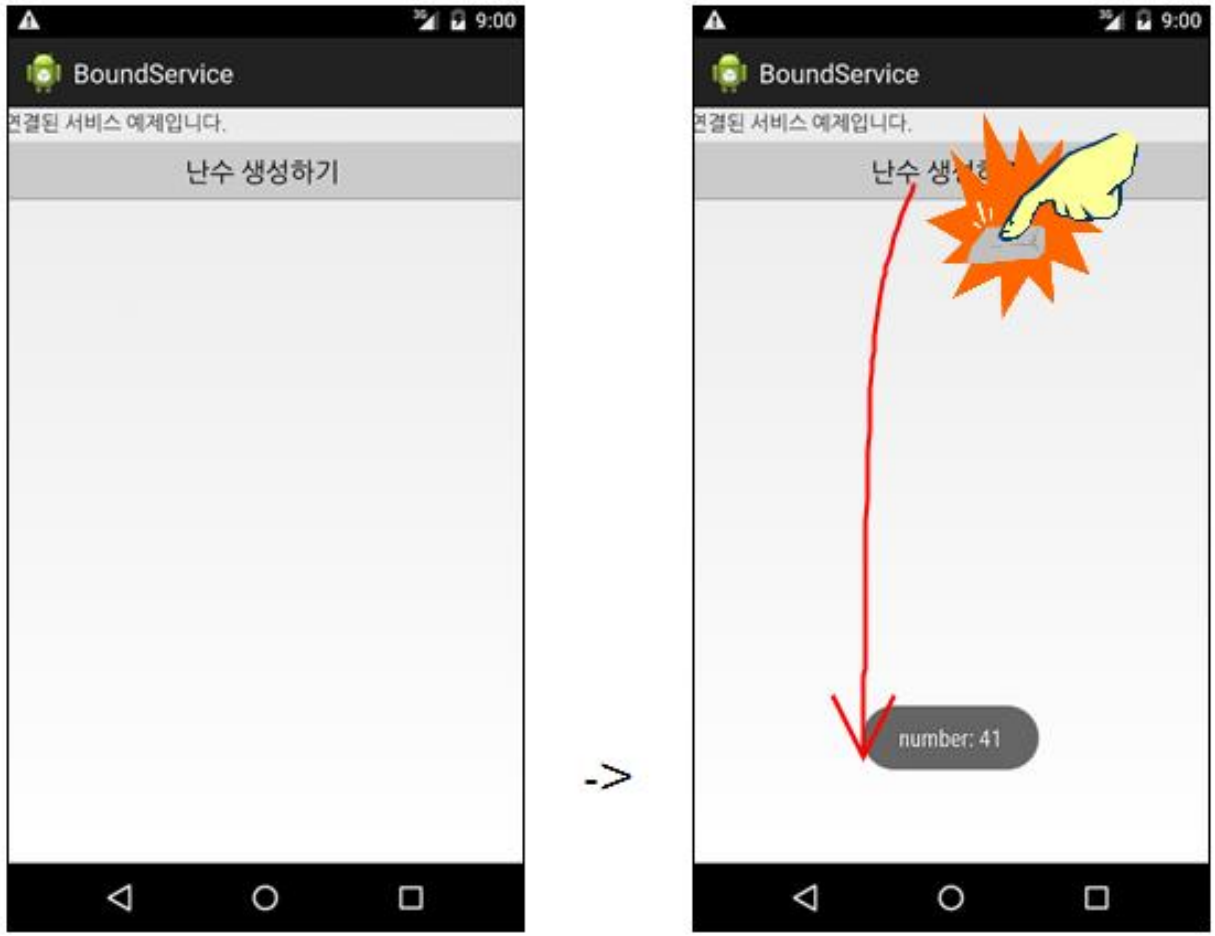
난수 발생 서비스 호출 코드 작성

```
public class MainActivity extends AppCompatActivity {
        LocalService mService;
        boolean mBound = false;

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

        @Override
        protected void onStart() {
                super.onStart();
                Intent intent = new Intent(this, LocalService.class);
                bindService(intent, mConnection,
Context.BIND_AUTO_CREATE);
        }

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

난수 발생 서비스 호출 코드 작성

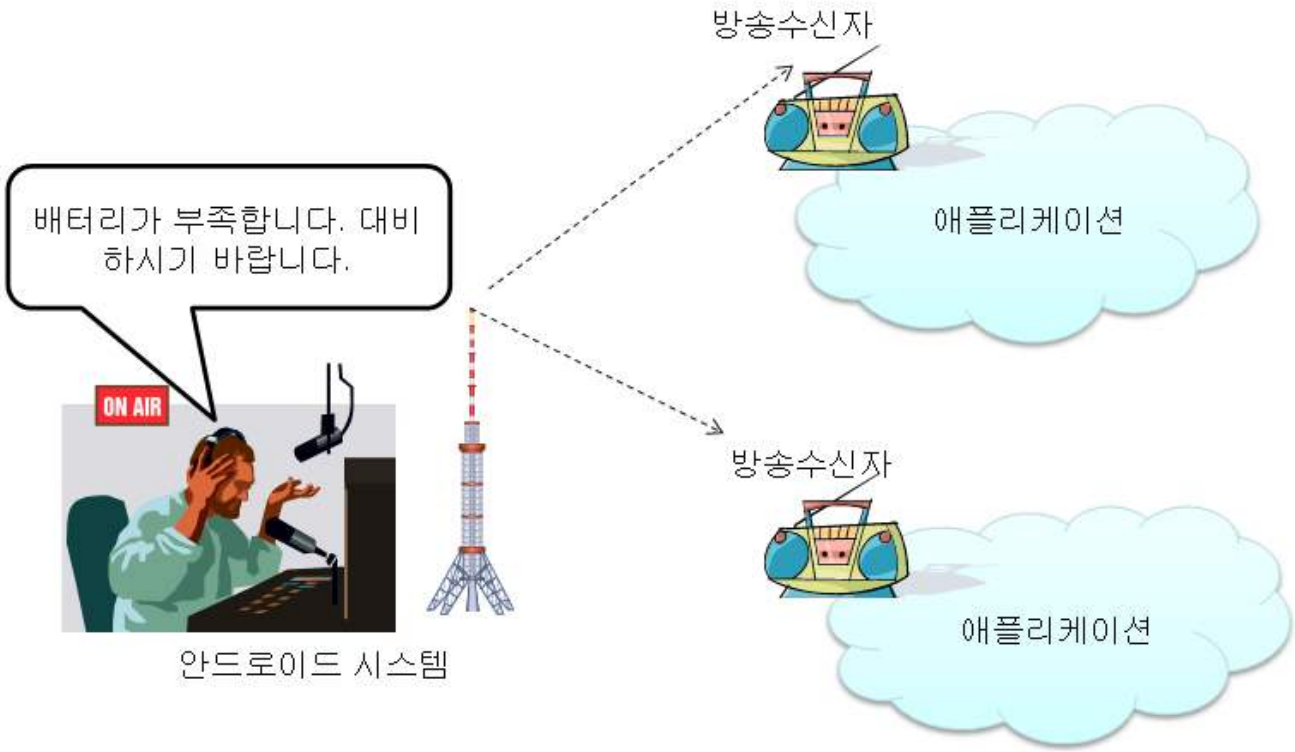
```
        // 버튼이 클릭되면 호출된다.
        public void onButtonClick(View v) {
                if (mBound) {
                        int num = mService.getRandomNumber();
                        Toast.makeText(this, "number: " + num,
Toast.LENGTH_SHORT).show();
                }
        }

        private ServiceConnection mConnection = new ServiceConnection() {
                // @Override
                public void onServiceConnected(ComponentName
className, IBinder service) {
                        LocalBinder binder = (LocalBinder) service;
                        mService = binder.getService();
                        mBound = true;
                }

                // @Override
                public void onServiceDisconnected(ComponentName arg0) {
                        mBound = false;
                }
        };
}
```

실행 결과

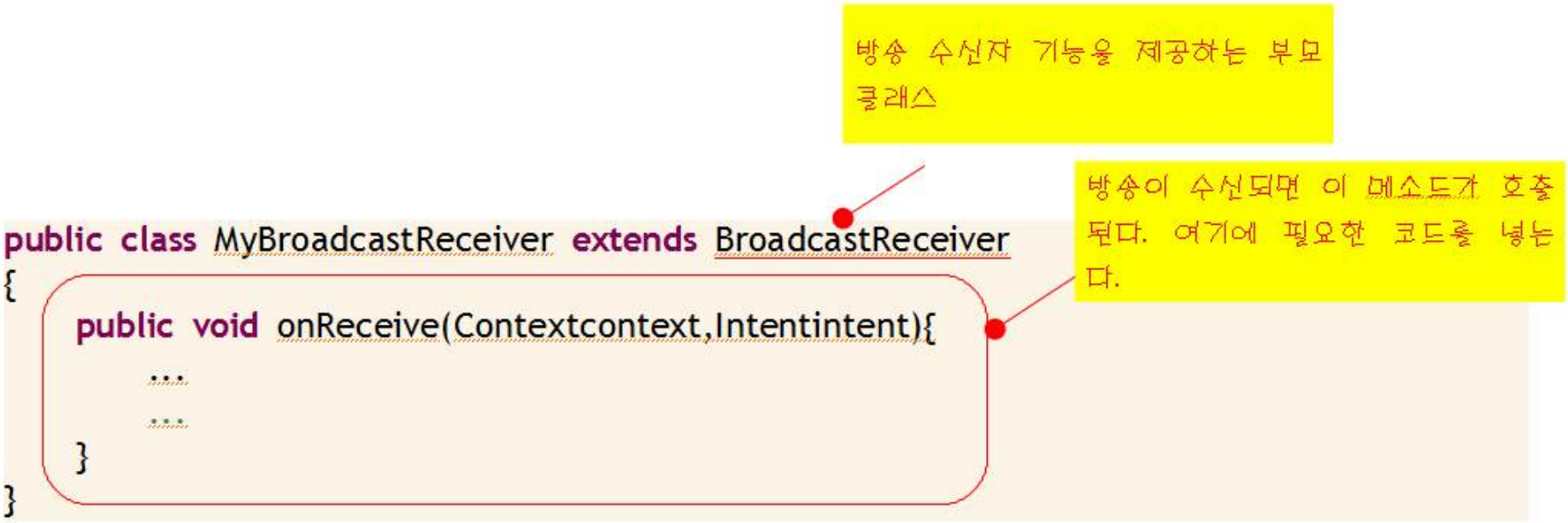

9:00
BoundService
연결된 서비스 예제입니다.
난수 생성하기
->
9:00
BoundService
연결된 서비스 예제입니다.
number: 41

방송 수신자

- 안드로이드 장치에서는 많은 이벤트들이 발생한다.
- 이벤트를 받는 컴포넌트가 방송 수신자이다.



방송 수신자의 구조


방송 수신자 기능을 제공하는 부모 클래스
방송이 수신되면 이 메소드가 호출된다. 여기에 필요한 코드를 넣는다.
public class MyBroadcastReceiver extends BroadcastReceiver
{
public void onReceive(Contextcontext,Intentintent){
...
...
}
}

중요한 방송

액션	설명
ACTION_TIME_TICK	1분마다 보내진다.
ACTION_TIME_CHANGED	현재 시각 설정
ACTION_TIMEZONE_CHANGED	시간대 변경
ACTION_BOOT_COMPLETED	부트 완료
ACTION_PACKAGE_ADDED	패키지 추가
ACTION_PACKAGE_CHANGED	패키지 변경
ACTION_PACKAGE_REMOVED	패키지 삭제
ACTION_MEDIA_MOUNTED	외부 저장 장치 마운트 완료
ACTION_MEDIA_REMOVED	외부 저장 장치 제거
ACTION_BATTERY_CHANGED	배터리 상태 변경
ACTION_BATTERY_LOW	배터리 저충전
ACTION_POWER_CONNECTED	전원 연결
ACTION_POWER_DISCONNECTED	전원 연결 해제
ACTION_SHUTDOWN	파워 오프

방송 수신자의 등록 방법

1. 매니페스트 파일에서 선언한다. - 최신 버전에서는 동작하지 않는다. 배터리 소모가 많기 때문이다. 여기서는 다루지 않는다.
2. 자바 코드에서 동적으로 등록한다. - 최신 버전에서는 이 방법을 사용하여야 한다. 예제에서 살펴보자.

방송 수신자의 인텐트 필터

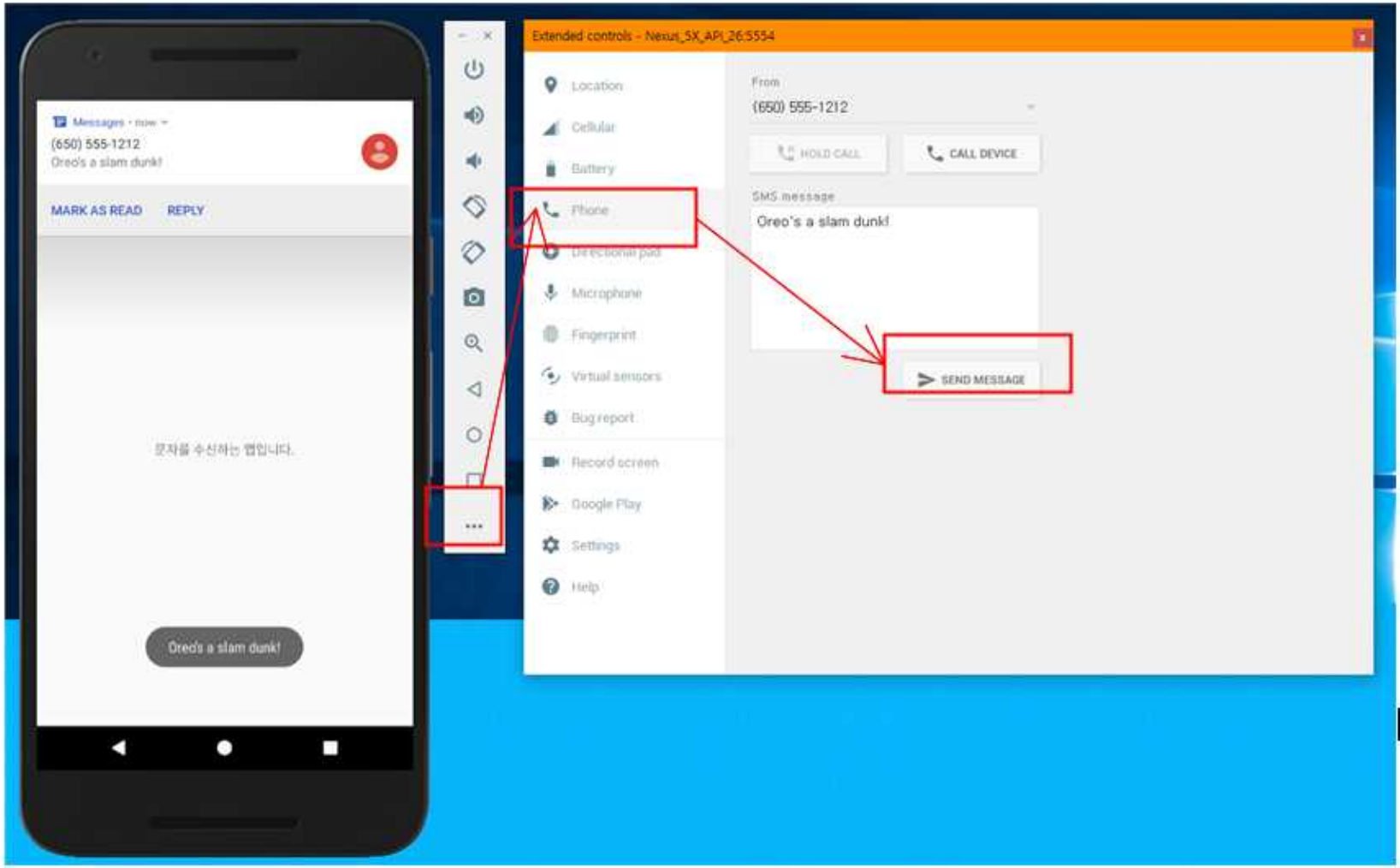
방송 수신자를 매니페스트 파일에 정의한다.

```
<receiver android:name="MyBroadcastReceiver">
    <intent-filter>
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
```

최신 버전에서의 변경 사항

- Android 8.0 (API 레벨 26)부터 대부분의 암시적 방송 수신자(앱을 구체적으로 타겟팅하지 않는 방송)의 경우, 매니페스트 파일에 선언하여도 동작하지 않는다. 방송 수신자는 반드시 registerReceiver()을 사용하여 코드에서 등록하여야 한다.

예제: 문자 메시지를 받는 방송 수신자



액티비티 정의

```
public class MainActivity extends AppCompatActivity {
    private int MY_PERMISSIONS_REQUEST_SMS_RECEIVE = 10;
    BroadcastReceiver receiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            if (intent.getAction().equals(Telephony.Sms.Intents.SMS_RECEIVED_ACTION)) {
                String smsSender = "";
                String smsBody = "";
                for (SmsMessage smsMessage :
Telephony.Sms.Intents.getMessagesFromIntent(intent)) {
                    smsBody += smsMessage.getMessageBody();
                }
                Toast.makeText(getApplicationContext(), smsBody,
Toast.LENGTH_SHORT).show();
            }
        }
    };
```

액티비티 정의

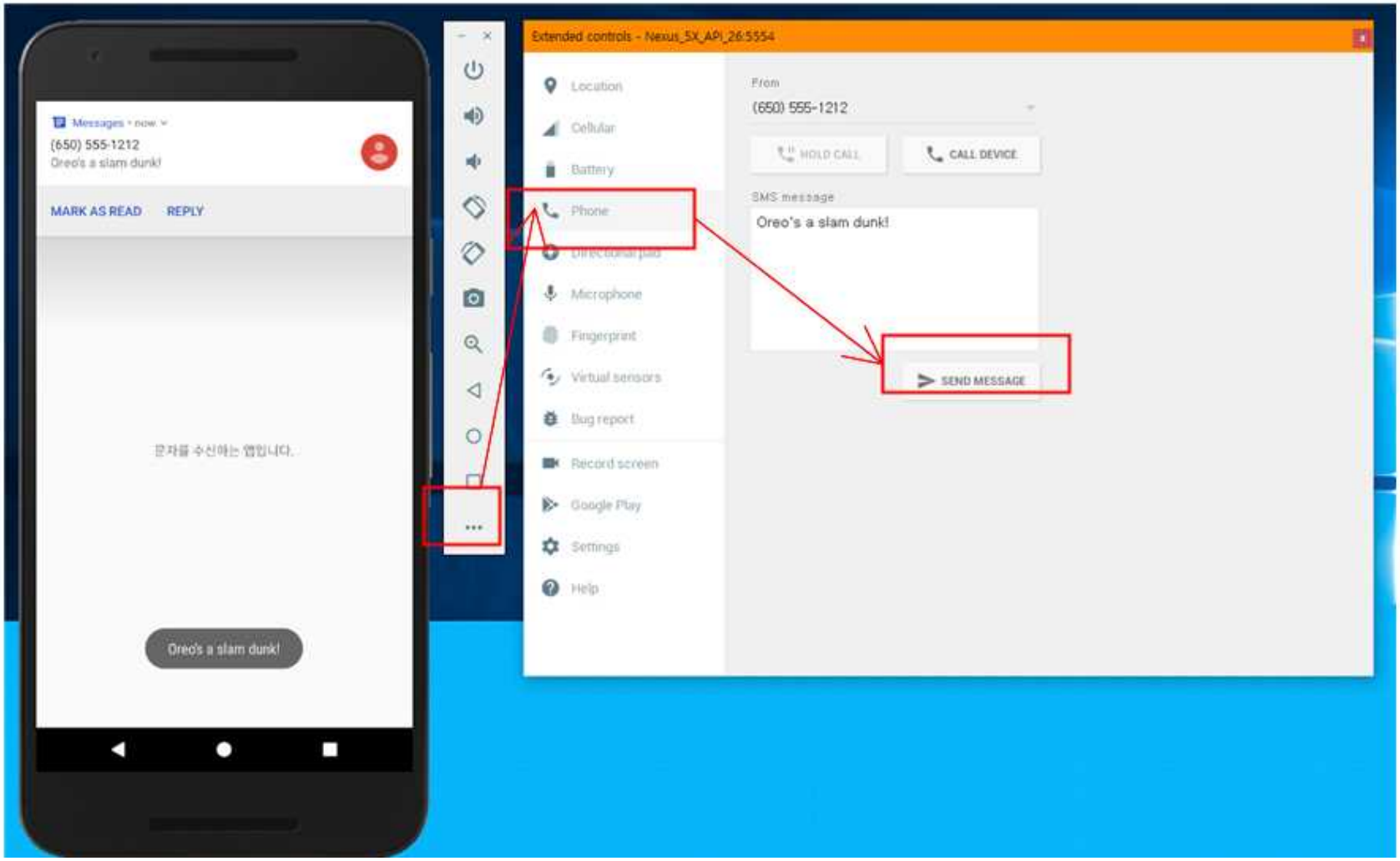
```
    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.RECEIVE_SMS},
                MY_PERMISSIONS_REQUEST_SMS_RECEIVE);
    }
    public void onResume() {
        super.onResume();
        IntentFilter filter = new IntentFilter();
        filter.addAction("android.provider.Telephony.SMS_RECEIVED");
        registerReceiver(receiver, filter);
    }
    public void onPause() {
        super.onPause();
        unregisterReceiver(receiver);
    }
}
```

메니페스트 파일

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/androia
    ...
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />

...
</manifest>
```

권한 요청


Messages • now
(650) 555-1212
Oreo's a slam dunk!
MARK AS READ
REPLY
문자를 수신하는 앱입니다.
Oreo's a slam dunk!
Extended controls - Nexus_5X_API_26:5554
Location
Cellular
Battery
Phone
Microphone
Fingerprint
Virtual sensors
Bug report
Record screen
Google Play
Settings
Help
From
(650) 555-1212
HOLD CALL
CALL DEVICE
SMS message
Oreo's a slam dunk!
SEND MESSAGE