

사용자가 스마트폰 화면을 보고 있지 않을 때 진동이나 소리를 울려서 앱의 상황을 알릴 수 있다.

- 진동 울리기: 먼저 퍼미션(permission)이 설정되어 있어야 한다.

```
// AndroidManifest.xml

<uses-permission android:name="android.permission.VIBRATE"/>
```

진동을 울리기 위해서는 API Level 31 버전부터는 VIBRATOR\_MANAGER\_SERVICE로 식별되는 VibratorManager을 이용해야 하고, 31 이전 버전에서는 Vibrator라는 객체, VIBRATOR\_SERVICE로 식별되는 SystemService를 이용하면 된다.

→ systemService는 시스템 레벨에서 제공하는 서비스이다. 안드로이드에서 서비스는 개발자가 직접 만들고 코드에서 직접 시작(start)해주어야 하지만, ServiceService는 일반 애플리케이션 서비스가 아닌 시스템에서 다양한 기능을 제공하기 위해 만들어 놓은 서비스로, 애플리케이션에서 이용할 때는 시작 작업 없이 getSystemService() 함수로 획득해서 이용하면 된다. 진동을 울리는 Vibrator가 대표적이며 이외에도 LayoutInflater, LocationManager, TelephonyManager 등의 다양한 SystemService를 제공한다.

```
Vibrator vibrator;
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
    VibratorManager manager = (VibratorManager) getSystemService(VIBRATOR_MANAGER_SERVICE);
    vibrator = manager.getDefaultVibrator();
} else {
    vibrator = (Vibrator) getSystemService(VIBRATOR_SERVICE);
}
}
```

진동을 울리기 위해서는 vibrate() 함수를 이용한다.

```
vibrator.vibrate(매개 변수);
```

API level 26 이전 버전에서 진동을 울리기 위해서는 vibrator() 함수의 매개변수로 millisecond, 진동이 울리는 시간을 사용한다. 1000로 설정하면 1초 동안 진동이 울리는 것이다. 진동을 여러 번 반복해서 울려야 할 때는 매개 변수가 2개인 함수를 사용한다.

```
public void vibrate(long milliseconds)
public void vibrate(long[] pattern, int repeat)
```

첫 번째 매개변수는 long형 배열로 배열 값의 홀수 번째 값이 대기 시간, 짝수 번째 값은 진동 시간을 의미한다. 두 번째 매개변수로는 진동을 이 패턴으로 얼마나 반복할 것인지를 설정한다. 0을 대입하면 무한 반복하겠다는 뜻으로, 코드에서 cancel() 함수로 취소해야 멈출 수 있다. 두 번째 매개변수를 -1로 지정하면 반복하지 않겠다는 뜻으로, 패턴대로 진동 알람이 한 번만 울리게 된다. 예를 들어, vibrator.vibrate(new long[]{500, 1000, 500, 1000}, -1); 이면, 진동 알람이 0.5초 쉬고 1초 울리고 0.5초 쉬고 1초 울리게 되고, 이 패턴의 반복 여부가 -1로 설정되어 있으므로 진동 알람은 반복 없이 한 번만 울리게 된다.

API Level 26 버전부터는 vibrator() 함수의 매개변수로 VibrationEffect 객체를 사용한다.

```
public void vibrate(android.os.VibrationEffect vibe)
```

VibrationEffect 객체는 createOneShot() 혹은 createWaveform() 함수로 만든다.

```
public static VibrationEffect createOneShot(long milliseconds, int amplitude)
public static VibrationEffect createWaveform(long[] timings, int[] amplitudes, int repeats)
```

createOneShot() 함수로 만든 VibrationEffect 객체는 한 번만 울리는 진동을 생성한다. 첫 번째 매개변수는 진동 시간이며 두 번째 매개변수는 진동 세기이다. 진동의 세기는 0~255 사이의 정수로 표현한다. 0으로 설정하면 진동이 울리지 않고, 255로 설정하면 스마트폰에서 지원하는 가장 강한 세기로 진동이 울린다. VibrationEffect.DEFAULT\_AMPLITUDE와 같은 상수로 지정하기도 하는데, 해당 상수를 사용하면 스마트폰의 기본 세기로 진동이 울린다.

createWaveform() 함수로 만든 VibrationEffect 객체로는 반복되는 진동을 생성한다. 첫 번째 매개변수는 진동이 울리는 시간의 패턴이며, 두 번째 매개변수는 진동의 세기 패턴이고, 세 번째 매개변수는 이 패턴의 반복 횟수이다.

```
// Ex. 사용자의 스마트폰 버전을 고려하여 진동을 한 번만 울리게 한다.
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
    vibrator.vibrate(VibrationEffect.createOneShot(1000, VibrationEffect.DEFAULT_AMPLITUDE));
} else {
    vibrator.vibrate(1000);
}
```

● 소리 울리기: 소리(beep)은 '짧은 알림음', '경적' 등을 나타내는 용어로 안드로이드 스마트폰에 내장된 기본 효과음을 이용하거나 직접 녹음한 효과음을 이용할 수 있다.

▶스마트폰에 등록되어 있는 효과음 사용하기: 안드로이드 시스템에는 간단한 알림, 알람, 전화 수신음 등이 있다. 이중 울리고자 하는 효과음을 선택하기 위해서는 스마트폰에 등록된 효과음의 식별자를 Uri 타입으로 획득한다. 효과음은 RingtoneManager의 getDefaultUri() 함수를 이용하여 획득하며 ALARM, NOTIFICATION, RINGTON 등 이 있다. Uri 값으로 식별되는 효과음을 재생할 수 있는 Ringtone을 얻어 play() 함수로 재생한다.

```
Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
Ringtone ringtone = RingtoneManager.getRingtone(getApplicationContext(), notification);
ringtone.play();
```

▶ 개발자 임의의 효과음: 녹음한 효과음을 리소스로 만들어서 사용한다. res 하위의 raw 폴더를 만들어서(res > New > Android Resource Directory > resource type: raw) 등록하면 되고, 이렇게 등록된 효과음은 MediaPlayer를 이용해서 간단하게 재생할 수 있다.

```
MediaPlayer player = MediaPlayer.create(this, R.raw.fallbackring);
player.start();
```

```

package com.myfamily.vibrationbeepractice;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.os.VibratorManager;
import android.view.View;

import com.myfamily.vibrationbeepractice.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {
    ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        binding.vibration.setOnClickListener(view -> {
            Vibrator vibrator;
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
                VibratorManager manager = (VibratorManager) getSystemService(VIBRATOR_M
                vibrator = manager.getDefaultVibrator();
            } else {
                vibrator = (Vibrator) getSystemService(VIBRATOR_SERVICE);
            }

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                vibrator.vibrate(VibrationEffect.createWaveform(new long[]{500,1000,500,
            } else {
                vibrator.vibrate(1000);
            }
        });

        binding.systemBeep.setOnClickListener(view -> {
            Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFI
            Ringtone ringtone = RingtoneManager.getRingtone(getApplicationContext(), not
            ringtone.play();
        });

        binding.customSound.setOnClickListener(view -> {
            MediaPlayer player = MediaPlayer.create(this, R.raw.asyoufadeaway);
            player.start();
        });

    }
}

```

