안드로이드 스튜디오 구조 파악하기

DEV 일기

2021/07/12 21:02 http://blog.naver.com/slovess1/222429114580

서버가 끝나고 이제 클라이언트 부분

본격 개발을 본격적으로 하기 전 안드로이드 스튜디오를 설치해

그 구조부터 파악하기로 한다(혼자 이해하기 위해 작성한 것이므로 정확하지 않을 수 있음!!)

https://developer.android.com/guide/topics/manifest/manifest-intro?hl=ko

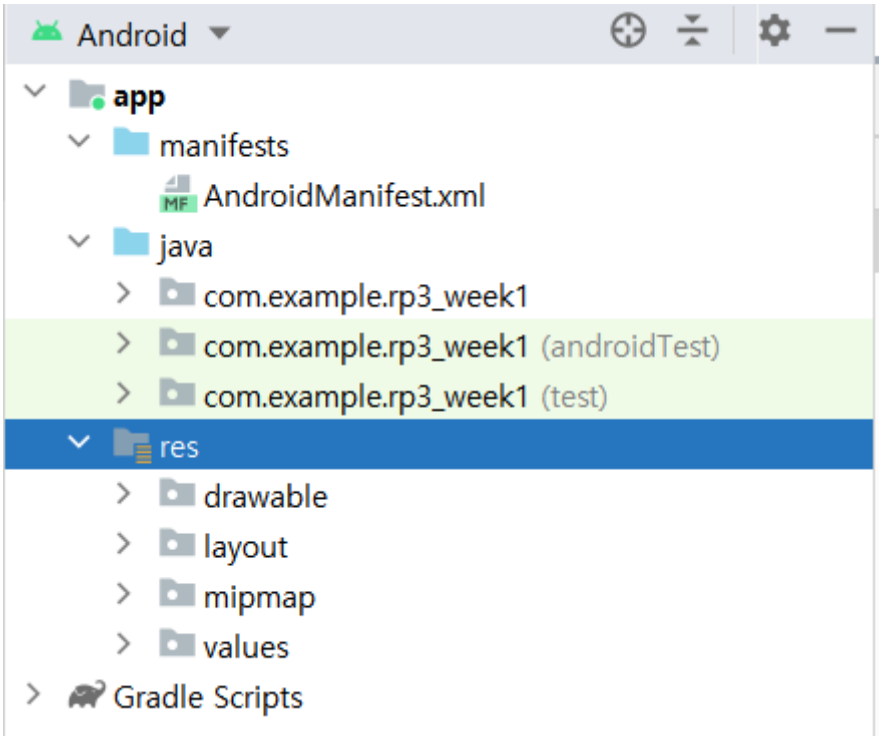
앱 매니페스트 개요 | Android 개발자 | Android Developers

Android 개발자 문서 가이드 앱 매니페스트 개요 목차 파일 기능 패키지 이름과 애플리케이션 ID 앱 구성 요소 권한 기기 호환성 모든 앱 프로젝트는 프로젝트 소스 세트 의 루트에 AndroidManifest.xml 파일(정확히 이 이름)이 있어야 합니다. 매니페스트 파일은 Android 빌드 도구, Android 운영체제 및 Google Play에 앱에 관한 필수 정보를 설명합니다. 매니페스트 파일은 다른 여러 가지도 설명하지만 특히 다음과 같은 내용을 선언해야 합니다. 앱의 패키지 이름(일반적으로 코드의 네임스페이스와 일치)...

developer.android.com

(developers 공식 홈페이지를 참고했다)

새로운 프로젝트를 생성하면 다음과 같은 파일 구조가 만들어진다



이렇게 app 안에 manifests, java, Gradle Scripts 디렉토리가 생성된다

manifests

manifests 안에는 AndroindManifest.xml 파일 하나가 들어있다

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.rp3_week1">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="RP3_WEEK1"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.RP3_WEEK1">
        <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>
```

안드로이드에서 만들 앱의 설정 값을 관리하는 공간이라 할 수 있는데,

구체적으로 앱의 기본 정보(앱 이름, 아이콘 이미지, 버전)와

앱을 실행하기 위한 필수 정보(앱의 파일 이름, 액티비티, 서비스, 권한, 테마 등)를 담고 있다

manifest는 앱의 신상명세라고 생각해도 무방할 듯 하다

코드를 하나씩 살펴보자

package="com.example.rp3_week1"

: 폴더의 경로 구조를 나타내고 있다

com/example에 rp_week1 프로젝트가 있다는 의미다

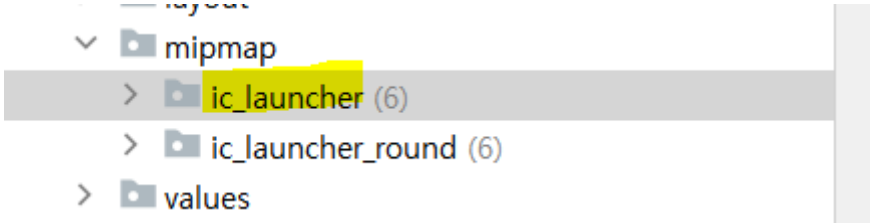
1. application

android:allowBackup="true"

: True로 설정돼 있기 때문에 백업 서비스를 이용하겠다는 것이다. 그럼 False일 땐 당연히 이용하지 않겠다는 뜻이겠지..?

찾아보니 구글 드라이브 계정을 통해 사용자 한 명당 25MB까지 자동 백업 기능을 실행할 수 있다고 한다

android:icon="@mipmap/ic_launcher"



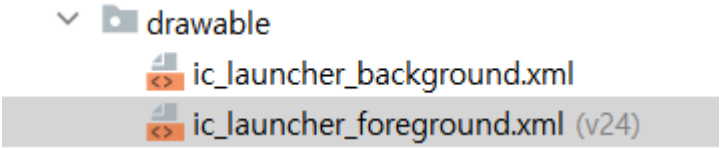
설치될 앱의 아이콘인데, res/mipmap 폴더 안에 ic_launcher 파일로 아이콘이 생성된다.

(@는 경로를 나타날 때 사용)

확인해보니 지금은 요로코롬 귀여운 안드로이드 이미지가 기본 설정돼 있다

```
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@drawable/ic_launcher_background" />
    <foreground android:drawable="@drawable/ic_launcher_foreground" />
</adaptive-icon>
```

여기서 background랑 foreground가 뭘 나타내는지 궁금해서



찾아봤더니 이미지 관련 drawble 폴더에서

background는 녹색 격자 배경이고, foreground는 안드로이드 머리인 것을 확인할 수 있었다

android:label="@string/app_name"

설치될 앱의 이름, res/values/strings.xml 파일에 들어있다

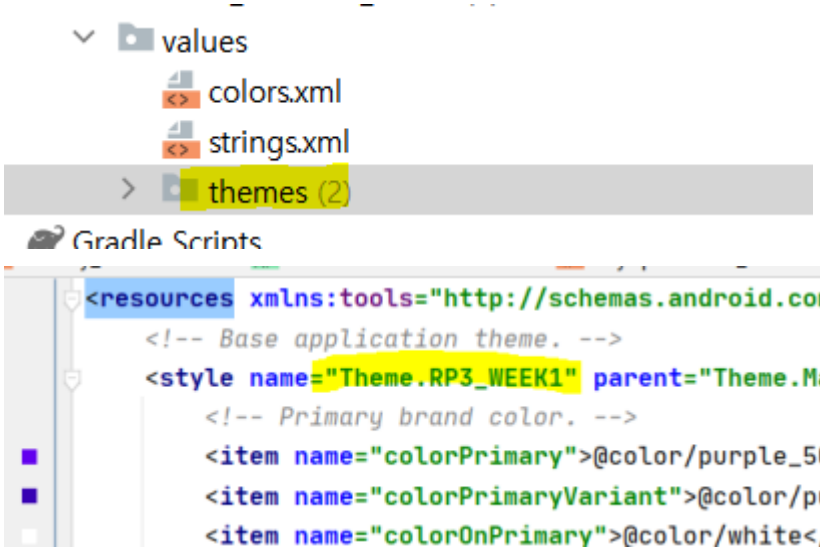
```
translations for all locales in the translations editor.
<resources>
    <string name="app_name">RP3_WEEK1</string>
</resources>
```

현재는 app_name으로 기본 지정돼 이씀

android:roundIcon="@mipmap/ic_launcher_round"
그냥 andorid:icon이면 사각형인데, 위와 같이 roundIcon일 땐 모서리가
둥근 아이콘을 설정하겠다는 뜻이다

android:supportsRtl="true"
일본이나 아랍 같은 곳에선 글자 읽는 법이 우리나라와는 반대인 경우가 있는데,
그것을 고려해 Right to Left layout 옵션을 True로 하겠다는 것이다

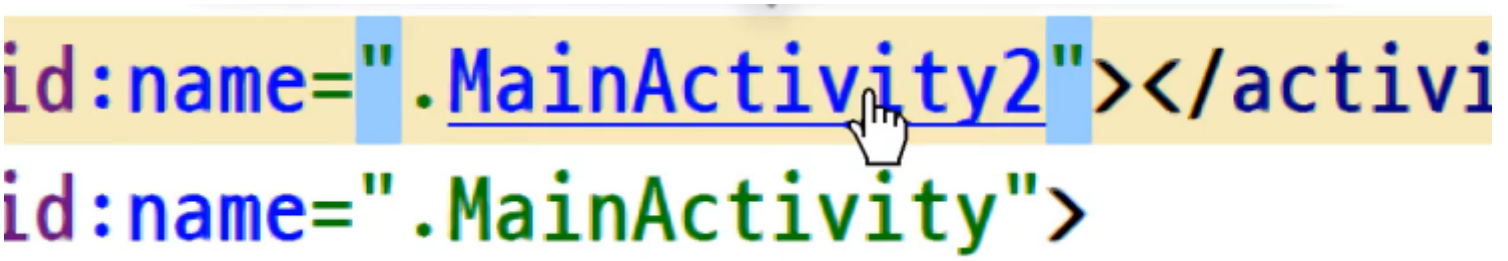
android:theme="@style/Theme.RP3_WEEK1"〉



Theme.RP3_WEEK1 테마가 적용돼 있다는 뜻

2. activity
〈activity android:name=".MainActivity"〉
activity는 앱 안의 단일 화면!
위 코드는 activity 사용에 대한 권한을 부여한다 생각하면 된다
새롭게 activity를 만들었으면 반드시 manifests에 해당 activity를 등록해줘야 한다는 사실!
그렇지 않으면 앱 실행시 죽음

activity는 복붙하기 보단 app 우클릭〉 new〉 Activity 해주는 게 편하다
왜냐구여? 복붙하면 일일이 manifests〉Andoridmanifest.xml에 따로 추가해줘야 하기 때문
클래스가 수백 개 있어도 여기에 반영돼 있지 않으면 안드로이드 시스템이 인지하지 못한다



따로 써주기 귀찮다면 맘 편하게 우클릭해서 새로 만들어주쟈..^^

우클릭해서 java에 해당 activity 클래스 파일이 만들어지면

res>layout에도 자동으로 xml 파일이 형성된다

3. intent-filter

activity가 실행될 때 필요한 정보를 정의하고, 이 정보를 바탕으로 동작 수행

주로 필터링하는 정보는 action, category, data가 존재

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

4. action

activity가 호출될 때 수행하는 동작이나, 특정 상태를 정의

<action android:name="android.intent.action.MAIN" />

위 코드에선 앱 실행시 처음 실행할 activity 지정 => .MainActivity가 첫 화면이라는 뜻

https://blog.naver.com/kkh0977/222223901529

41. (AndroidStudio/android/java) intent-filter 설명 및 주요 속성인 action, category, data 설명

/* ==========&#x3D...

blog.naver.com

(이 블로그 참고)

- <action android:name="android.intent.action.MAIN" /> : 처음으로 실행되는 액티비티로 설정 *: 위에서 설명했던 기본 설정돼 있는 코드*
- <action android:name="android.intent.action.VIEW" /> : URL 로 호출되는 액티비티로 설정
- <action android:name="android.intent.action.DEFAULT" /> : action.VIEW 와 동일
- <action android:name="android.intent.action.EDIT" /> : 수정을 수행 하기 위한 액티비티로 설정
- <action android:name="android.intent.action.DELETE" /> : 삭제를 수행 하기 위한 액티비티로 설정
- <action android:name="android.intent.action.DIAL" /> : 전화걸기를 수행 하기 위한 액티비티로 설정
- <action android:name="android.intent.action.CALL" /> : 전화걸기를 수행 하기 위한 액티비티로 설정
- <action android:name="android.intent.action.SENDTO" /> : 이메일을 보내기 위한 액티비티로 설정
- <action android:name="android.intent.action.ANSWER" /> : 전화 착신을 하기 위한 액티비티로 설정
- <action android:name="android.nfc.action.NDEF_DISCOVERED"/> : NFC 통신을 수행하기 위한 액티비티로 설정
- <action android:name="android.nfc.action.WEB_SEARCH"/> : 웹 검색을 수행하기 위한 액티비티로 설정
- <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE"/> : NFC 양방향 통신을 수행하기 위한 액티비티로 설정

```
<activity android:name=".MainActivity2">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

이 intent-filter 위치에 따라 어떤 화면이 제일 먼저 나오는지 결정됨

위처럼 써주면 .MainActivity2가 첫 화면으로 등장하게 된다

5. category

수행할 동작, 상태에 대한 추가적인 정보를 정의하는 category

<category android:name="android.intent.category.LAUNCHER" />

그럼 코드 속 LAUNCHER는 뭘까

해당 activity에 대한 아이콘이 화면에 표시돼야 한다는 뜻이다

- <category android:name="android.intent.category.LAUNCHER"/> : 사용자 앱에 설치된 목록에 보여지게 한다 *:위*

~~*에서 설명했던 기본 설정돼 있는 코드*~~

- <category android:name="android.intent.category.DEFAULT"/> : 암시적 인텐트를 받을 수 있게 한다
- <category android:name="android.intent.category.BROWSABLE"/> : 웹브라우저 기능을 포함한다, 링크로 연결된 콘텐츠(이미지, 웹문서, 이메일 메시지 ...)를 보여줄 수 있게 한다

+ data

코드엔 없는 data!

Intent가 수행될 때 필요한 항목, 타입을 지정해 일치하는지 확인 작업을 진행

- <data android:host="test_host" android:scheme="test_schema"/> : 인텐트에 접근하려면 test_host와 test_schema로 접근해야 한다
- <data android:host="test_host" android:scheme="test_schema"/> : 인텐트에 접근하려면 test_host와 test_schema로 접근해야 한다
- <data android:mimeType="text/plain" /> : 인텐트에 접근하려면 text/plain 형식으로 접근해야 한다

6. Intent

그렇다면 근본적인 Intent 개념은 무엇일까

Intent는 화면 이동의 베이스,

즉 앱 내부에서 진행 방향을 알려주는 역할이다

(공식 문서를 봐도 잘 모르겠어서 다른 블로그랑 유튜브 찾아봐서 이해)

- **액티비티 시작**

  `Activity`는 앱 안의 단일 화면을 나타냅니다. `Activity`의 새 인스턴스를 시작하려면 `Intent`를 `startActivity()`로 전달하면 됩니다. `Intent`는 시작할 액티비티를 설명하고 모든 필수 데이터를 담습니다.

  액티비티가 완료되었을 때 결과를 수신하려면, `startActivityForResult()`를 호출합니다. 액티비티는 해당 결과를 이 액티비티의 `onActivityResult()` 콜백에서 별도의 `Intent` 객체로 수신합니다. 자세한 내용은 액티비티 가이드를 참조하세요.

- **서비스 시작**

  `Service`는 사용자 인터페이스 없이 백그라운드에서 작업을 수행하는 구성 요소입니다. Android 5.0(API 레벨 21) 이상부터는 `JobScheduler`로 서비스를 시작할 수 있습니다. `JobScheduler`에 대한 자세한 내용은 `API-reference documentation`을 참조하세요.

  Android 5.0(API 레벨 21) 이하 버전은 `Service` 클래스의 메서드를 사용하면 서비스를 시작할 수 있습니다. 서비스를 시작하여 일회성 작업을 수행하도록 하려면(예: 파일 다운로드) `Intent`를 `startService()`에 전달하면 됩니다. `Intent`는 시작할 서비스를 설명하고 모든 필수 데이터를 담고 있습니다.

  서비스가 클라이언트-서버 인터페이스로 디자인된 경우, 다른 구성 요소로부터 서비스에 바인딩하려면 `Intent`를 `bindService()`에 전달하면 됩니다. 자세한 내용은 서비스 가이드를 참조하세요.

- **브로드캐스트 전달**

  브로드캐스트는 모든 앱이 수신할 수 있는 메시지입니다. 시스템은 시스템이 부팅될 때 또는 기기가 충전을 시작할 때 등 시스템 이벤트에 대한 다양한 브로드캐스트를 전달합니다. `Intent`를 `sendBroadcast()` 또는 `sendOrderedBroadcast()`에 전달하면 다른 앱에 브로드캐스트를 전달할 수 있습니다.

내가 이해하기론 activity끼리 직접적인 연결을 할 수 없기에

intent를 통해 이 둘을 연결시켜주는 느낌이다

인텐트의 파라미터로 activity 클래스를 전달하면

A 화면에서 B 화면으로 바뀌는

그런 역할을 하는 거라고 이해했는데 맞겠지..?

공식 문서에 의하면 intent는 두 가지 유형이 있다

1) 명시적 인텐트(Explicit Intent)

한 마디로 확실한 거! intent에 특정 이름을 지정해 호출할 대상이 확실할 경우 사용된다

=> 특정 component나 activity가 명확하게 실행돼야 할 경우 사용

2) 암시적 인텐트(Implicit Intent)

불확실한 거! intent의 액션과 데이터를 지정하긴 했지만, 호출 대상이 달라질 수 있는 경우 사용된다

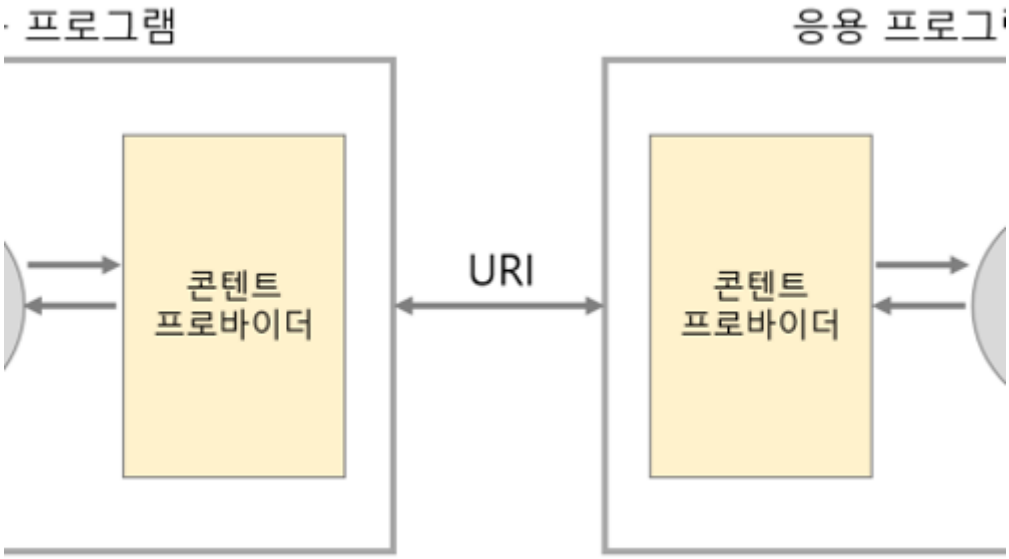
어떤 타입과 액션을 하는지 특징으로 추론해 컴포넌트 간 통신을 가능하게 만듦

ex) 친구가 나한테 문서 파일을 카톡으로 보냈는데, 그걸 클릭하면 열 수 있는 여러 앱들이 목록에 뜨잖아여? 굿노트, 메모장, 폴라리스오피스 등 이런 과정을 가능하게 하는 것이 이 암시적 인텐트라고 생각하면 쉬움

안드로이드의 4대 컴포넌트

4개의 컴포넌트는 하나씩 독립적인 형태이며, Intent로 상호 통신이 이뤄짐

https://blog.naver.com/hhayoung96/222298217453



[Android] 안드로이드 4대 컴포넌트에 대해 알아보자

#android #4대컴포넌트 #액티비티 #서비스 #브로드캐스트리시버 #콘텐트프로바이더 ■ 안드로이드 4대 컴포...

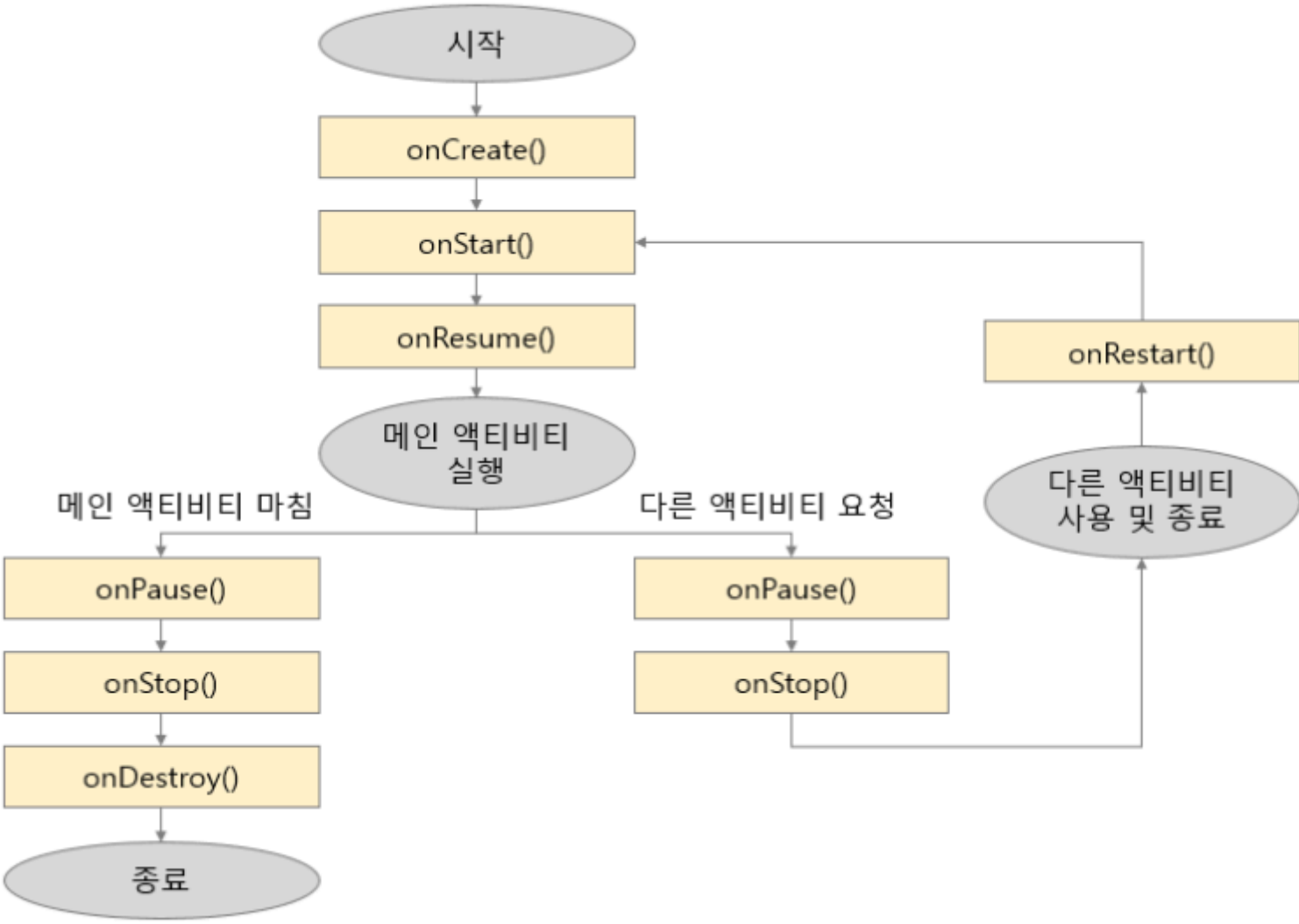
blog.naver.com

(이 블로그 참고)

1) Activity

사용자에게 보여지는 UI 화면

위에서 manifest에 지정해줬던 activity = 단일 화면 알져?



액티비티 생명주기(Life Cycle)

2) BroadCastReceiver

안드로이드 단말기에서 발생하는 이벤트, 정보를 전달 받고 처리하는 컴포넌트

ex) 배터리 부족, WIFI 설정

이벤트가 발생하면 바로 바로 처리해야 하기 때문에

언제나 수신 받을 준비를 하고 있음

3) Service

사용자에게 보여지지 않는 백그라운드의 프로세스 역할

모든 서비스는 서비스 클래스를 상속받아 이뤄짐

ex) 우리가 컴퓨터를 하면 V3, 알약 같은 프로그램이 보이진 않지만 계에에에ㅔ속 실행되죠?

그리고 음악을 들으면서 다른 작업을 한다던지 이런 것들 모두 백그라운드에서 동작하는 서비스의 역할이라 할 수 있음!!

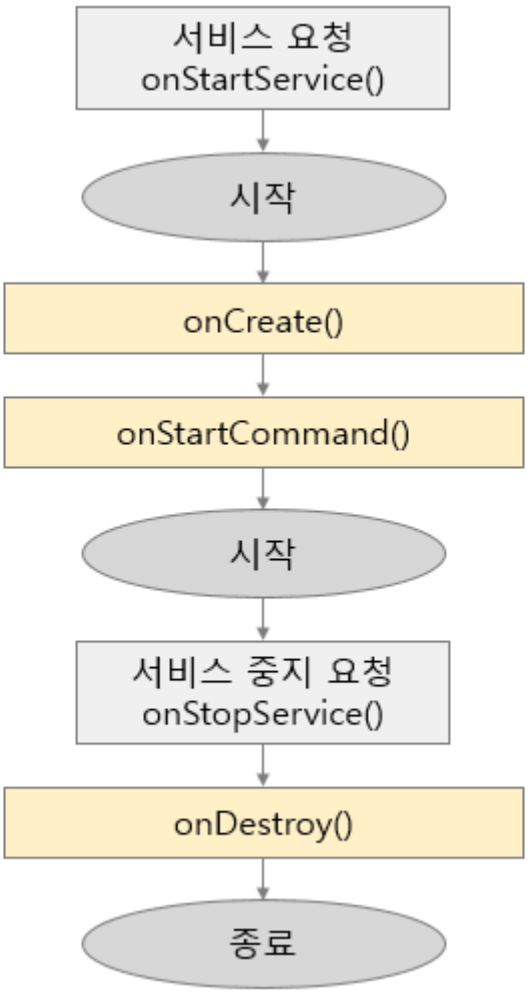
구현 방식

- StartService

한 프로세스 안에서 컴포넌트 간 유기적 역할 수행

- BindService

여러 프로세스 간 유기적 역할 수행



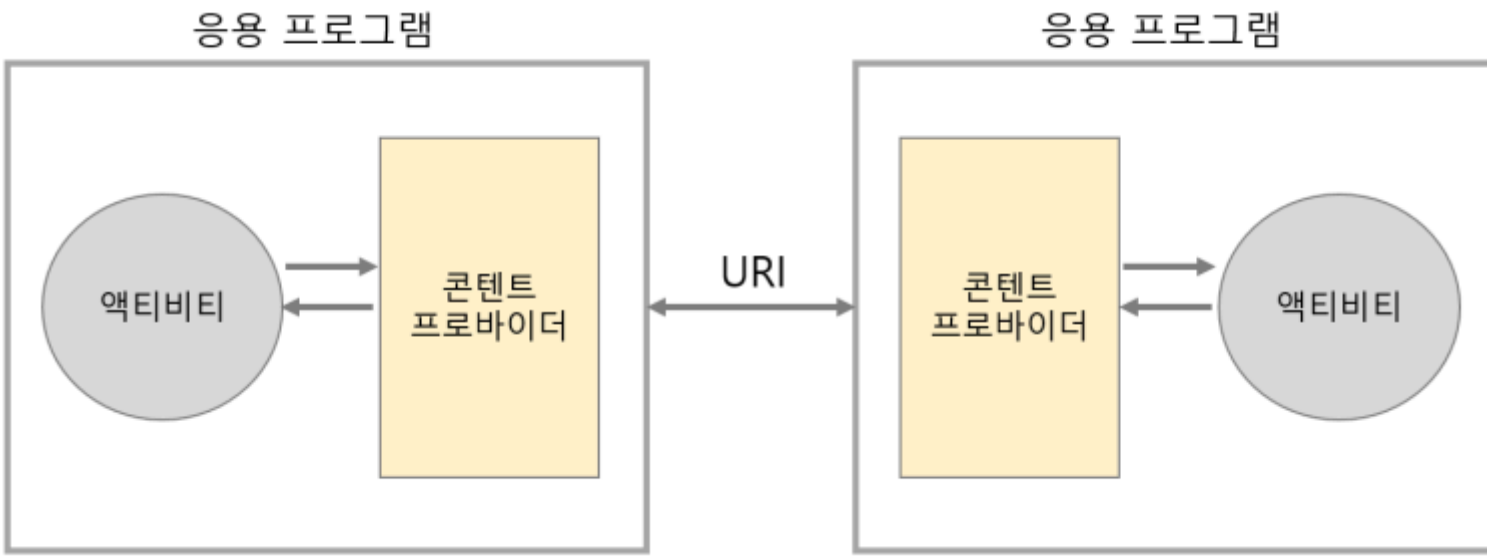
서비스 생명주기(Life Cycle)

4) Content provider

앱 사이에서 데이터를 주고 받을 수 있게 해줌

정보를 제공하는 방법으론 URI(Uniform Resource Identifier)가 있는데,

URI는 Content provider가 제공하는 데이터에 접근하기 위한 주소라 생각하면 된다



개념도

연락처 전화번호	android.provider.Contacts.Phones.CONTENT_URI
통화 기록	android.provider.CallLog.Calls.CONTENT_URI
브라우저 북마크	android.provider.Browser.BOOKMARKS_URI
브라우저 검색 기록	android.provider.Browser.SEARCHES_URI
시스템 설정 값	android.provider.System.CONTENT_URI
내장 미디어의 이미지	android.provider.MediaStore.Image.Media.INTERNAL_CONTENT_URI
내장 미디어의 동영상	android.provider.MediaStore.Video.Media.INTERNAL_CONTENT_URI
내장 미디어의 오디오	android.provider.MediaStore.Audio.Media.INTERNAL_CONTENT_URI
외장 미디어의 이미지	android.provider.MediaStore.Image.Media.EXTERNAL_CONTENT_URI
외장 미디어의 동영상	android.provider.MediaStore.Video.Media.EXTERNAL_CONTENT_URI
외장 미디어의 오디오	android.provider.MediaStore.Audio.Media.EXTERNAL_CONTENT_URI

안드로이드 자체에서 제공하는 Content provider