인텐트란 무엇인가요 ?

채넛

목차

인텐트란 ?

다른 앱 컴포넌트에 요청을 보내는 메시징 객체

* 컴포넌트 : (Activity, Service, Broadcast receiver, Content Provider)

인텐트 유형 명시적 인텐트

컴포넌트 이름을 명시해서 사용하는 방식의 인텐트

자신의 앱 내에서 컴포넌트를 시작하기 위해 주로 사용한다.

사용 예시

1. 앱 내에서 새로운 액티비티를 시작
2. 백그라운드에서 파일을 다운로드 하기 위해 서비스를 시작

인텐트 유형 명시적 인텐트

코드 예시

```
val intent = Intent(this, ToIntentActivity().javaClass)
    .putExtra("info", "정보 전송")
startActivity(intent)
```

인텐트 유형 암시적 인텐트

특정 컴포넌트 이름을 지정하지 않고 수행할 동작을 선언하는 방식의 인텐트

action, category, data 기반으로 어떤 컴포넌트가 인텐트를 처리할 수 있는 지 결정한다.

인텐트 유형 암시적 인텐트

Action의 종류

1. ACTION_VIEW : 사용자에게 보여줄 때 사용 (웹페이지, 연락처, 사진, ...)
2. ACTION_SEND : 다른 앱으로 공유할 때 사용 (이미지, 텍스트 전송, ...)
3. ACTION_DIAL : 다이얼 앱을 열어 전화번호 입력한 상태로 표시
4. ACTION_CALL : 지정된 번호로 바로 전화 걸 때 사용 (사용 시, 권한 필요)
5. ACTION_EDIT : 수정할 때 사용 (연락처 수정, ...)
6. ACTION_SENDTO : 특정 수신자에게 데이터 전송할 때 사용 (이메일, SMS, ...)

인텐트 유형 암시적 인텐트

Category의 종류

1. DEFAULT : 대부분 암시적 인텐트에서 필수, 명시하지 않으면 연결되지 않는다.
2. BROWSABLE : 웹 브라우저 등에서 URI를 통해 액티비티를 실행할 수 있게 한다.
3. LAUNCHER : 앱 런처에서 실행 가능한 액티비티임을 명시한다.
4. HOME : 홈 화면 역할의 액티비티임을 명시한다.
5. ALTERNATIVE : 대체 동작을 제공하는 컴포넌트에 사용한다.
6. SELECTED_ALTERNATIVE : 선택된 대체 동작에 사용한다.

인텐트 유형 암시적 인텐트

Data의 종류

1. mimeType : 처리 가능한 MIME 타입을 지정
2. scheme : URI의 scheme 지정
3. host : URI의 host 지정
4. port : URI의 port 지정
5. path : URI의 경로 지정
6. pathPattern : 와일드카드 패턴으로 경로 지정
7. pathPrefix : 경로의 접두어 지정

인텐트 유형 암시적 인텐트

송신측 예시

```kotlin
// 전화 다이얼러 열고 번호 입력
val dialIntent = Intent(Intent.ACTION_DIAL, "tel:010-1234-5678".toUri())
startActivity(dialIntent)

// 지도 앱에서 위치 보기
val mapIntent = Intent(Intent.ACTION_VIEW, "geo:37.565350,127.01445".toUri())
startActivity(mapIntent)

// 이메일 보내기
val emailIntent = Intent().apply {
        action = Intent.ACTION_SENDTO
    "mailto:someone@email.com".toUri().also { data = it }
    putExtra(Intent.EXTRA_SUBJECT, "제목")
    putExtra(Intent.EXTRA_TEXT, "내용")
}
startActivity(emailIntent)

// uri 해당하는 페이지로 이동
val intent = Intent(Intent.ACTION_VIEW)
intent.data =
    "https://github.com/woowacourse-study/2025-Manifest-Android-Interview-Study".toUri()
startActivity(intent)
```

인텐트 유형 암시적 인텐트

수신측 예시

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data android:scheme="http" android:host="www.example.com" />
    <data android:mimeType="image/*" />
</intent-filter>
```

사용 예시 Activity

송신측

```
val intent = Intent(this, ToIntentActivity().javaClass)
                .putExtra("info", "정보 전송")
startActivity(intent)
```

수신측

```
val info = intent.getStringExtra("info")
```

사용 예시 Activity

수신측 메서드

```
public int getIntExtra(String name, int defaultValue) {
    return mExtras =수= null ? defaultValue :
        mExtras.getInt(name, defaultValue);
}

public @Nullable String getStringExtra(String name) {
    return mExtras == null ? null : mExtras.getString(name);
}
```

기본형 타입은 null 값을 가질 수 없기에 동일한 이름의 번들 객체가 없다면 기본값을 설정하도록 메서드 지원

참조형 타입은 null 값을 가질 수 있기에 기본값을 제공하지 않고 null 값을 반환하도록 메서드 지원

IntentService

Service에서 인텐트 기반 작업을 쉽게 처리하기 위해 주로 사용되었던 객체

하나의 Worker thread에서 처리한다.

앱의 메인 루프를 막지는 않지만 한 번에 하나의 요청만 처리한다.

Android 8.0 (API26, 오레오) 부터 모든 백그라운드 실행 제한의 영향을 받아
Android 11(API30) 부터 Deprecated 되었다.

cf. Service : 사용자 인터페이스 없이 백그라운드에서 작업을 수행하는 구성요소

cf. Worker thread : 특정한 작업(시간이 오래 걸리는 연산, 네트워크 통신 등)을 Main thread와 분리하여 처리하기 위해 만들어진 스레드로 UI, Main thread가 바쁘지 않게 백그라운드에서 무거운 작업을 처리하는 역할을 한다.

IntentService

IntentService의 사용 대안

Job Api가 Service를 대체하기 위해 나왔다.

JobScheduler : API 21 이상에서만 사용 가능하다. 다만, 일부 기기에서 버그 및 제한이 존재한다.

➔ JobDispatcher : 구글 플레이 서비스 의존성, 일관성이 부족하다.

➔ JobIntentService : 즉시 실행이 보장되지 않거나, 제조사 커스텀 OS에서 동작이 불안정하다.

➔ WorkManager

IntentService

WorkManager

Android Jetpack 라이브러리

백그라운드 작업 관리 시스템으로, 앱이 종료되거나 기기가 재부팅되어도 반드시 실행되어야 하는 작업을 안전하게 예약하고 실행할 수 있다.

내부적으로 API 레벨에 따라 JobScheduler, AlarmManager, BroadcastReceiver 등을 알맞게 선택해준다.

백그라운드에서 주기적으로 서버와 앱의 데이터를 동기화하거나, 백그라운드에서 즉시 실행할 작업 예약에 사용된다.

Doze mode 같은 절전 기능을 지키고 있어서 개발자가 신경쓸 부분을 줄여준다.

질문

❓ 명시적 인텐트와 암시적 인텐트의 차이점은 무엇이며, 각각 어떤 시나리오에서 사용해야 하나요?

실행할 컴포넌트를 정확히 명시하느냐 아니냐의 차이가 있다.

명시적 인텐트의 경우 앱의 화면전환에 사용할 수 있다.
예를 들어 채팅이라는 서비스를 이용할 때 채팅방을 클릭해서 채팅방 내부로 이동하는 경우 명시적 인텐트를 사용할 수 있다.

암시적 인텐트의 경우 작업의 의도만 전달하여 일치하는 컴포넌트가 실행되도록 할 때 사용할 수 있다.
예를 들어 사용자가 링크를 클릭할 경우 웹 브라우저로 이동하는 경우 암시적 인텐트를 사용할 수 있다.
또 다른 경우로는 번호에 암시적 인텐트를 적용해 사용자 기기의 다이얼에 번호를 입력해주는 경우가 있다.

암시적 인텐트 사용 시 주의 사항으로는 개인정보 민감 작업에 대해서는 제한 될 수 있다.

차이점에 대해 추가적으로 말하면 명시적은 자신의 앱 내부 컴포넌트를 호출할 때만 사용가능하고,
암시적 인텐트는 다른 앱의 컴포넌트도 호출할 수 있다.

질문

❓안드로이드 시스템은 암시적 인텐트를 처리할 앱을 어떻게 결정하며, 적합한 애플리케이션을 찾지 못하면 어떻게 되나요?

설치된 모든 앱의 인텐트 필터를 확인하여, 해당 인텐트를 처리할 수 있는 컴포넌트를 찾는다.
적합한 인텐트 필터가 여러 개라면, 사용자에게 선택화면을 보여주며 원하는 앱을 선택할 수 있다.

만약, 적합한 애플리케이션을 하나도 찾지 못하면 ActivityNotFoundException 예외가 발생하며, 앱이 강제 종료될 수 있다.
이를 방지하기 위해, Intent.resolveActivity()로 사전에 처리 가능한 앱이 있는 지 확인하는 것을 권장한다.

```
val intent = Intent(Intent.ACTION_VIEW)
intent.data = "https://github.com/woowacourse-study/2025-Manifest-Android-Interview-Study".toUri()
if (intent.resolveActivity(packageManager) != null) {
    startActivity(intent)
} else {
    // 처리할 앱이 없는 경우의 예외 처리
}
```

참고

https://leanpub.com/manifest-android-interview-kr

https://developer.android.com/guide/components/intents-filters?hl=en

https://developer.android.com/develop/background-work/background-tasks/persistent?hl=en