

인텐트, 서비스



시작하면서



◎ 목차

- 인텐트
- 서비스
- 알림
- 방송 수신자
- 알람



인텐트(Intent)

- ◎ '의도' 또는 '의향'이라는 뜻
- ◎ 일종의 메시지 전달 메커니즘
 - 컴포넌트간 의사 소통하는 수단
 - 예)액션으로 Intent.ACTION_VIEW를 포함하는 인텐트인 경우:
 - 다른 컴포넌트에게 무엇인가 보여주는 처리를 요청함
 - 독립적인 컴포넌트들을 서로 연결된 하나의 시스템으로 구성해 주는 효과

- ◎ 구성 요소
 - 액션: 수행될 액션, 획득한 정보나 보고될 행동을 나타내는 문자열
 - ACTION_CALL, ACTION_VIEW, ACTION_MAIN(주 액티비티 실행)
 - 데이터: 데이터에 대한 URI(Uniform Resource Identifier)
 - 통화하고자 하는 전화번호인 경우: <tel:URI>
 - 연락처인 경우: content//contacts/
 - 컴포넌트: 인텐트를 받아서 처리할 컴포넌트의 명시적 패키지와 클래스 지정
 - 유형: 명시적인 MIME 형식 지정
 - 범주: 인텐트에 대한 추가 메타 데이터
 - LAUNCHER로 분류되면 런처에 표시
 - 기타: 처리할 대상에게 보낼 추가적인 정보



인텐트(Intent)

◎ 인텐트 필터(filter)

- 특정 의도를 포함하는 인텐트를 처리하는 존재
- 컴포넌트는 intent filter를 통해 다른 컴포넌트의 액션 요청을 처리할 수 있다
고 안드로이드에게 알림
- AndroidManifest.xml에 선언

◎ 인텐트 경로배정(routing)

- 인텐트와 인텐트 필터를 서로 연결 해주는 과정
- 해당 인텐트에 가장 잘 부합하는 인텐트 필터를 갖는 유일한 컴포넌트를 찾음
- 종류
 - 명시적 인텐트
 - 해당 인텐트를 전달해야 하는 수신처 컴포넌트의 명칭을 명시하는 방식
 - 동일 애플리케이션내 타 액티비티 호출시 사용
 - 암시적 인텐트
 - 수신처를 명시하지 않은 경우 가장 적합한 컴포넌트를 찾아야 함
 - 안드로이드가 제공하는 기능 사용시 정확하게 알지 않고서도 사용 가능
 - `Intent i=new Intent(Intent.ACTION_DIAL,Uri.parse("tel:123456789"));`
 - `startActivity(i);`
 - 전화번호가 입력된 다이얼용 액티비티가 호출되고, 전화용 애플리케이션이 실행됨
 - 암시적 인텐트에서 사용되는 대표적인 액션 종류: <표 8-1>

암시적 인텐트(Intent)



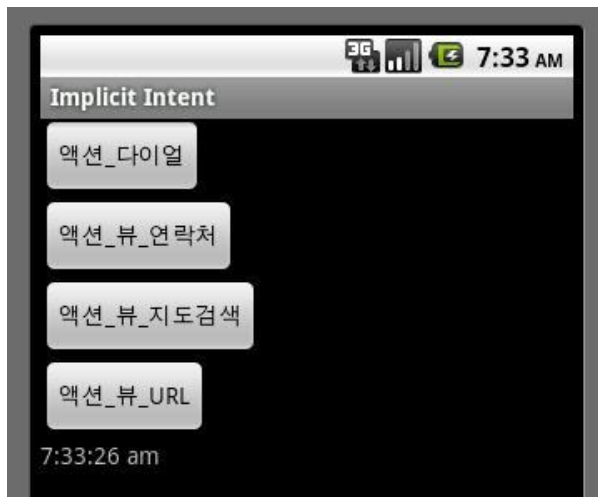
- 사례
 - 전화 걸기 액션
 - `Intent i = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:027305800"));`
 - `startActivity(i);`
 - 연락처 보기 액션
 - `Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("content://contacts/people"));`
 - `startActivity(i);`
 - 위치 검색 액션
 - `Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("geo:37.56,126.89")); // 경복궁`
 - `startActivity(i);`
 - 인터넷 검색 액션
 - `Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com/"));`
 - `startActivity(i);`

암시적 인텐트(Intent)



◎ <실습 8-1> 암시적 인텐트 호출

- IntentImplicitDemo 프로젝트 생성
 - <코드 8-1>를 main.xml에 복사
 - <코드 8-2>~<코드 8-3>을 IntentImplicitDemo.java에 복사
 - 특기사항: 위치 검색에서 지도 사용하므로 Google API로 컴파일/실행
 - 참고: 구글 지도에서 오른쪽 상단의 링크(link)를 클릭하면 지점의 위도/경도 좌표를 획득



암시적 인텐트(Intent)



명시적 인텐트



- 하나의 응용프로그램 안에서 다른 컴포넌트를 호출하여 실행하는 방식
- 인텐트를 통해 실행시킬 액티비티는 AndroidManifest 파일 내에 액티비티 클래스명을 선언
- 호출 액티비티는 자신이 설정한 인텐트를 통해 실행된 액티비티로부터 결괏값의 반환 가능

◎ 사례

- 반환값이 없는 액티비티 호출
 - 주 액티비티에서 인텐트를 통해 AnotherActivity를 호출
 - `Intent intent = new Intent(getApplicationContext(), AnotherActivity.class);`
 - `startActivity(intent);`

명시적 인텐트



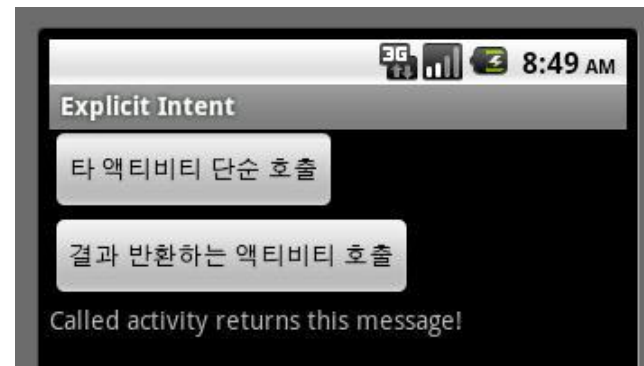
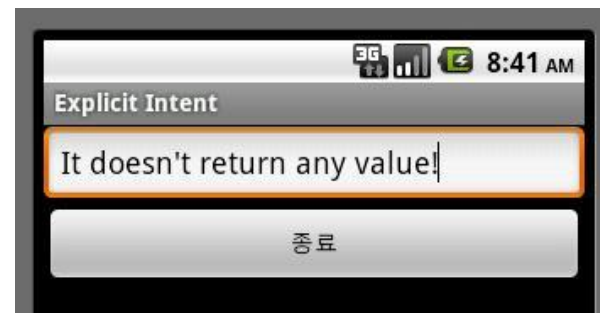
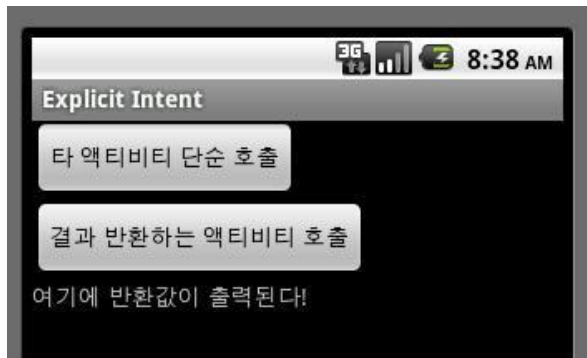
- 반환값을 돌려받는 애플리케이션
 - 주 액티비티에서 인텐트를 통해 `AnotherActivity`를 호출
 - `Intent intent = new Intent(getApplicationContext(), AnotherActivity.class);`
 - `startActivityForResult(intent, ANOTHER_ACTIVITY);`
 - 호출된 액티비티: `AnotherActivity`
 - `Intent i = new Intent();`
 - `i.putExtra(TEXT_RESULT, result);`
 - `setResult(RESULT_OK, i);`
 - 피호출 액티비티로부터 반환시 주 액티비티에서 호출되는 콜백 메소드
 - `onActivityResult()`
 - 액티비티간 호출/반환 관계: [그림 8-3]

명시적 인텐트



◎ <실습 8-2> 명시적 인텐트 호출

- IntentExplicitDemo 프로젝트 생성
 - <코드 8-4>를 main.xml에 복사
 - <코드 8-5>를 launched_view.xml에 복사
 - <코드 8-6>을 AndroidManifest.xml에 수정
 - <코드 8-7> ~ <코드 8-11>을 IntentExplicitDemo.java에 복사



서비스(Service)



◎ 기능

- 액티비티와 연결되지 않고 백그라운드로 작업 수행
- 휴대폰내 IPC(Interprocess Communication)를 위한 원격 호출 인터페이스 제공
- 사용할 서비스를 AndroidManifest 파일에 선언
 - `<service`
 - `android:name="MyService"`
 - `android:label="MyService" />`
- 서비스 속성: <표 8-2>

◎ GUI없이 실행하는 형태

예)MP3 재생기, 스포트 점수 모니터

예)위치 관리자, 미디어 컨트롤러, 알림 관리자 등

서비스(Service)



⊙ 서비스 실행 개시/중단 메소드

- `startService(new Intent(this, MyService.class));`
- `stopService(i)`

⊙ 서비스의 생애주기

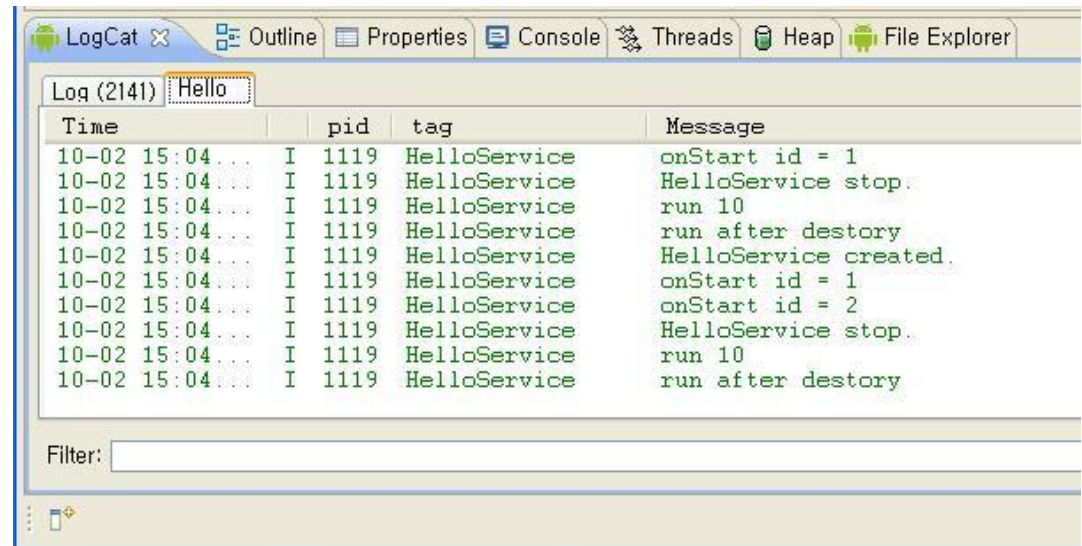
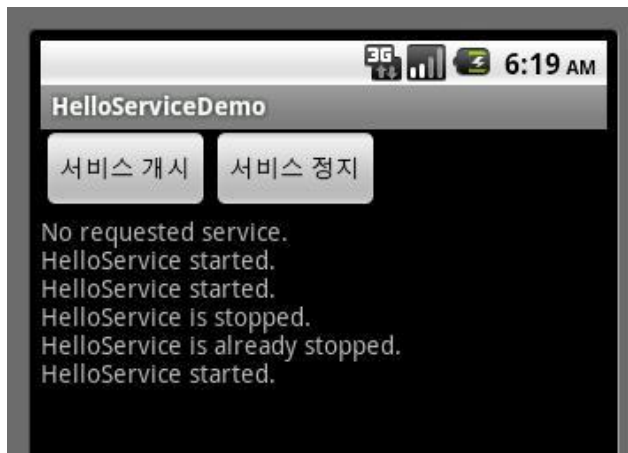
- `onCreate()`
- `onStart()`
- 중간단계
 - 백그라운드 데몬 : `onStartCommand()`
 - 원격 호출 인터페이스: `onBind()`
- `onDestroy()`

서비스(Service)



◎ <실습 8-3> 서비스 호출

- 서비스를 생성하고 중단하는 실습
- ServiceDemo 프로젝트 생성
 - <코드 8-12>를 main.xml에 복사
 - <코드 8-13>~<코드 8-14>을 ServiceDemo.java에 복사, <코드 8-15>~<코드 8-16>을 MyService.java에 복사
 - 프로젝트를 실행하고 [그림 8-4](a)와 같이 서비스 개시와 정지 버튼을 교대로 누름
 - LogCat에 서비스 실행/중단 과정이 출력: [그림 8-4](b)



알림(Notification)



- 백그라운드 프로세스가 사용자와 통신하는 방법 중의 하나
- 종류
 - Toast Notification
 - Status bar notification
 - Dialog notification

상태바 알림

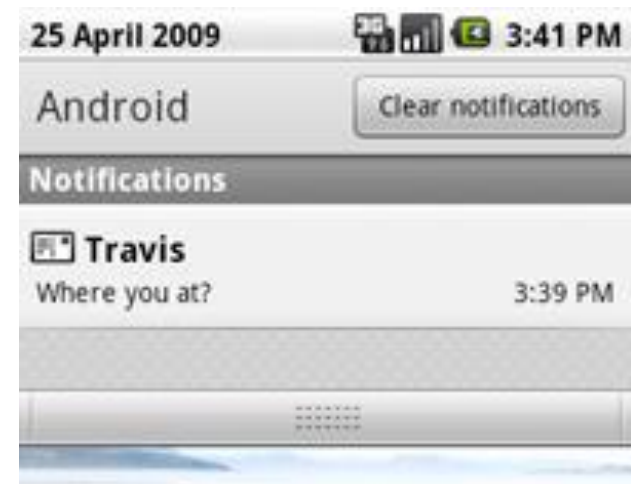
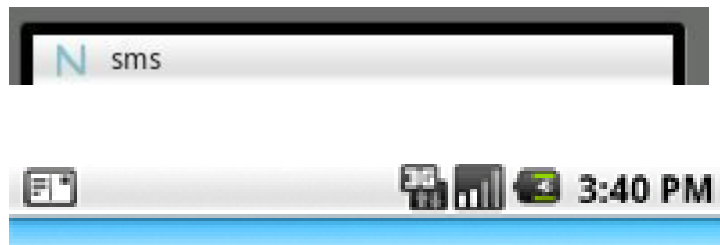


◎ 상태바 알림

- 사용자가 확인할 때까지 상태바에 아이콘으로 계속 표시
- 상태바의 아이콘을 드래그하면 확장 메시지가 나타남
- 확장 메시지를 선택하면 미리 설정된 인텐트 PendingIntent가 활성화
- 관련 컴포넌트가 실행

◎ 실행 예

- SMS 문자 수신 알림



상태바 알림(Notification)



◎ 생성자

- Notification(int icon, CharSequence tickerText, long when);
 - 첫번째 매개변수: 알림에 사용할 아이콘
 - 두번째 매개변수: 상태바에 아이콘이 처음 나타날 때 잠시 출력될 짧은 문자열
 - 세번째 매개변수: 알림이 발생한 시간을 지정
 - System.currentTimeMillis 메소드에서 얻은 현재시간 지정

방송 수신자(Broadcast Receiver)

◎ 방송(broadcast):

- 시스템에 변화가 생겼을 때 이를 알려주는 방식
- 애플리케이션들이 항상 방송에 주의를 기울이고 있으면 휴대 단말에서 발생하는 여러 가지 변화에 대한 제어가 가능
- 서비스나 알림과 비슷한 부류
- 방송을 수신하는 애플리케이션은 브로드캐스트 리시버를 갖고 있고 방송 수신을 대기
 - 이것은 사용자와 직접 상호작용하지 않음

◎ 방송 이벤트를 위한 인텐트

- 인텐트의 다른 용도
- 메시지를 컴포넌트들에게 익명으로 방송
 - sendBroadcast 메소드
 - 예) 배터리 충전 수준, 네트워크 연결, 걸려오는 전화 등에 대응 가능
- Broadcast Receiver의 제한조건
 - onReceiver 메소드는 약10초내에 완료해야 함
 - 매우 짧고 특별한 라이프 사이클을 가짐

방송 수신자(Broadcast Receiver)

◎ 절차

- 브로드캐스트 리시버를 구현
- 브로드캐스트 리시버를 AndroidManifest.xml에 등록
- 시스템 서비스 사용 권한을 AndroidManifest.xml에 등록

◎ <실습 8-4> 방송 수신자

- Booting시 방송되는 BOOT_COMPLETED 이벤트를 받음
- BRDemo 프로젝트를 생성
 - <코드 8-19>를 BRDemo.java에 복사
 - <코드 8-20>를 AndroidManifest.xml에 복사
 - 에뮬레이터를 다시 부팅하여 onReceive() 메소드가 호출되는 것을 확인



알람(Alarm)

- ◎ 미리 지정한 시간에 이벤트를 발생하는 시스템 서비스
 - 알람은 OS가 관리
 - 애플리케이션이 종료되어도 발생
 - 알람 취소하는 경우
 - 사용자가 취소
 - 단말기가 부팅

 - AlarmManagerDemo.java
 - `AlarmManager am = (AlarmManager) getSystemService(ALARM_SERVICE);`
 - `Intent intent = new Intent(AlarmManagerDemo.this, AlarmService.class);`
 - `PendingIntent.getBroadcast(AlarmManagerDemo.this, 0, intent, 0);`
 - AlarmService.java
 - `public class AlarmService extends BroadcastReceiver { ...`
 - `onReceive(Context context, Intent intent) ...}`

알람(Alarm)



◎ <실습 8-5> 알람 설정

- AlarmManagerDemo 프로젝트 생성
 - <코드 8-21>을 main.xml에 복사
 - <코드 8-22>를 AlarmManagerDemo.java에 복사
 - <코드 8-23>을 AlarmService.java에 복사

