

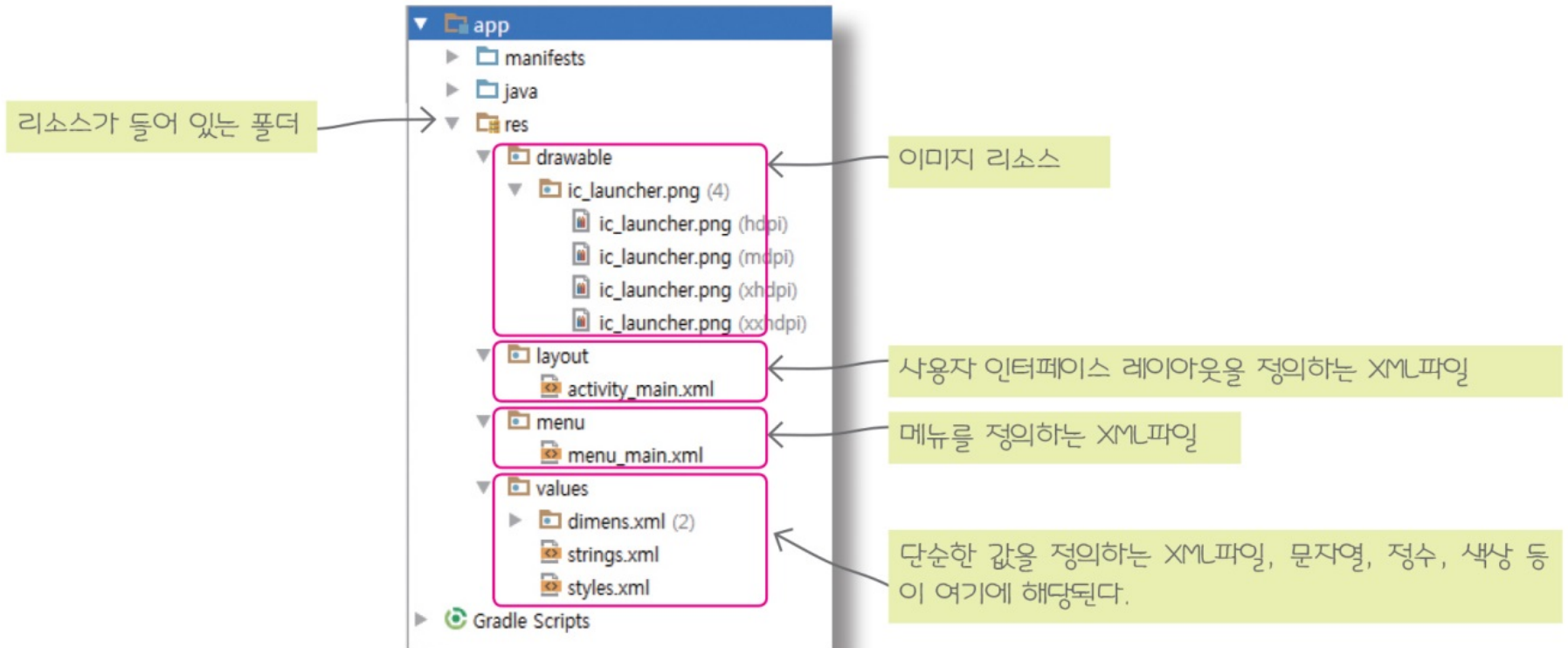
CHAP 11. 리소스와 보안

리소스

- 리소스(resource)란 이미지, 문자열, 레이아웃, 동영상 파일 등을 의미한다.
- 리소스는 특별하게 이름 지어진 리소스 디렉토리에 모여 있어야 한다.



리소스의 예



리소스의 종류

디렉터리	리소스 타입
animator/	프로퍼티 애니메이션을 정의하는 XML 파일
anim/	트윈 애니메이션을 정의하는 XML 파일
color/	컬러의 상태 리스트를 정의하는 XML 파일
drawable/	비트맵 파일(.png, .9.png, .jpg, .gif)이나 다음과 같은 리소스 타입으로 컴파일되는 XML 파일 <ul style="list-style-type: none">• Bitmap 파일• Nine-Patches (가변-크기 bitmaps)• State 리스트• Shapes• Animation drawables
layout/	사용자 인터페이스 레이아웃을 정의하는 XML 파일
mipmap/	여러 가지 해상도의 애플리케이션 아이콘 파일이 저장된다.
menu/	애플리케이션 메뉴를 정의하는 XML 파일
raw/	시스템에 의하여 압축되지 않는 원본 파일. 이들 리소스는 Resources.openRawResource()를 R.raw.filename의 형태의 아이디로 호출하여 사용한다. 하지만 만약 원본 파일 이름이나 파일 계층구조를 접근하려고 한다면 assets/ 디렉터리에 저장하는 것도 좋다. assets/에 있는 파일은 리소스 아이디가 없으므로 AssetManager를 통해서만이 읽을 수 있다.

리소스의 종류

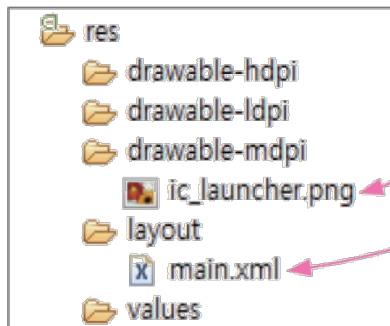
values/	<p>단순한 값을 정의하는 XML 파일, 문자열, 정수, 색상 등이 여기에 해당된다. <resources> 요소 안에 여러 가지 자식 요소를 둘 수 있다. 예를 들어서 <string> 요소는 R.string 리소스를 생성하고 <color> 요소는 R.color 리소스를 생성한다.</p> <p>각 리소스는 각자의 XML 요소 안에 정의되므로 파일의 이름은 중요치 않다. 하지만 다음과 같이 서로 다른 리소스를 별도의 파일에 두는 것도 좋다.</p> <ul style="list-style-type: none">• arrays.xml(배열)• colors.xml(색상)• dimens.xml(크기)• strings.xml(문자열)• styles.xml(스타일)
xml/	<p>실행 시간에 Resources.getXML()을 호출하여서 읽을 수 있는 XML파일,</p>

리소스 참조

- R 클래스

R.java

```
public final class R {  
    ...  
    public static final class drawable {  
        public static final int ic_launcher=0x7f020000;  
    }  
    ...  
    public static final class layout {  
        public static final int main=0x7f030000;  
    }  
    public static final class string {  
        public static final int app_name=0x7f040000;  
        ...  
    }  
}
```



참조 방법

- 코드에서 참조

리소스 타입

리소스 이름

R.string.hello

- XML에서 참조

리소스 타입

리소스 이름

@string/hello

기본 리소스와 대체 리소스

- **기본 리소스(default resource)**

- 장치 구성과 상관없이 기본적으로 사용되는 리소스이다.

- **대체 리소스(alternative resource)**

- 대체 리소스는 단말기의 특성에 맞게 선택되는 리소스이다.

기본 리소스와 대체 리소스



(a) 기본 리소스만 있는 경우, 2개의 장치에 동일한 레이아웃이 사용된다.



(b) 기본 리소스와 대체 리소스가 있는 경우, 장치의 크기에 따라 서로 다른 레이아웃이 사용된다.

대체 리소스를 제공하는 방법

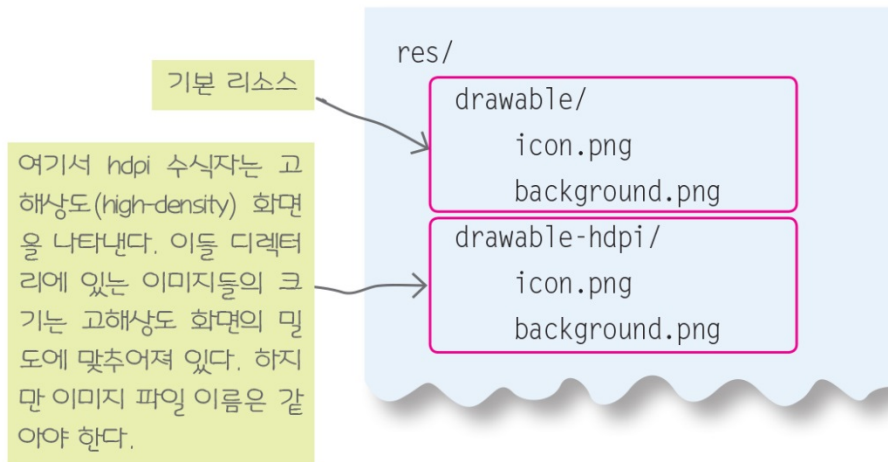
- 기본 디렉토리 이름에 특정한 장치 구성의 이름을 붙인 디렉토리에 리소스들이 저장

디렉토리 이름 형식

<resource_name> - <config_qualifier>

예)

res/drawable/
res/drawable-hdpi/
res/layout/
res/layout-land/



리소스 수식자

: 유효한 대체 리소스

수식자	값	설명
MCC와 MNC	예: mcc310 mcc310-mnc004	mobile country code(MCC), mobile network code(MNC) 예를 들어서 mcc310은 U.S., mcc310-mnc004 is U.S.의 버라이즌
언어 구분	예: en en-rUS	언어 뒤에 2글자로 된 ISO 639-1 언어 코드를 붙인다.
스크린 크기	small normal large xlarge	small: QVGA 스크린 normal: 전통적인 중밀도 HVGA 스크린 large: 중밀도 VGA 스크린
스크린 종횡비	long notlong	long: 세로로 긴 스크린 notlong: 가로로 긴 스크린
스크린 방향	port land	port: 세로(portrait) 방향 land: 가로(landscape) 방향

리소스 수식자

: 유효한 대체 리소스

UI 모드	car desk television	car: 자동차에서 표시 desk: 책상 위에서 표시 television: 텔레비전에서 표시
밤 모드	night not night	night: 밤 모드 notnight: 낮 모드
스크린 픽셀 밀도(dpi)	ldpi mdpi hdpi xhdpi xxhdpi xxxhdpi	ldpi: 저밀도 스크린; 약 120dpi. mdpi: 중밀도 스크린, 약 160dpi. hdpi: 고밀도 스크린; 약 240dpi. xhdpi: 고밀도 스크린; 약 320dpi. xxhdpi: 고밀도 스크린; 약 480dpi. xxxhdpi: 고밀도 스크린; 약 640dpi.
키보드 여부	keysexposed keyshidden keysoft	keysexposed: 장치가 하드웨어 키보드를 가지고 있고 노출되어 있음 keyshidden: 장치가 하드웨어 키보드를 가지고 있으나 감추어져 있음 keysoft: 장치가 소프트웨어 키보드를 가지고 있음

리소스 탐색 과정 #1

- 리소스

```
drawable/  
drawable-en/  
drawable-fr-rCA/  
drawable-en-port/  
drawable-en-notouch-12key/  
drawable-port-1dpi/  
drawable-port-notouch-12key/
```

- 장치구성

```
Locale = en-GB  
Screen orientation = port  
Screen pixel density = hdpi  
Touchscreen type = notouch  
Primary text input method = 12key
```

리소스 탐색 과정 #2

• 리소스

- ① Locale = en-GB
- ② Screen orientation = port
Screen pixel density = hdpi
Touchscreen type = notouch
Primary text input method = 12key

①

```
drawable/  
drawable-en/  
drawable-fr-rCA/  
drawable-en-port/  
drawable-en-notouch-12key/  
drawable-port-ldpi/ ←  
drawable-port-notouch-12key/
```

장치는 hdpi이지만 drawable-port-ldpi/가 삭제되지 않는다. 스크린 밀도는 이 시점에서는 모두 일치되는 것으로 간주된다.

①

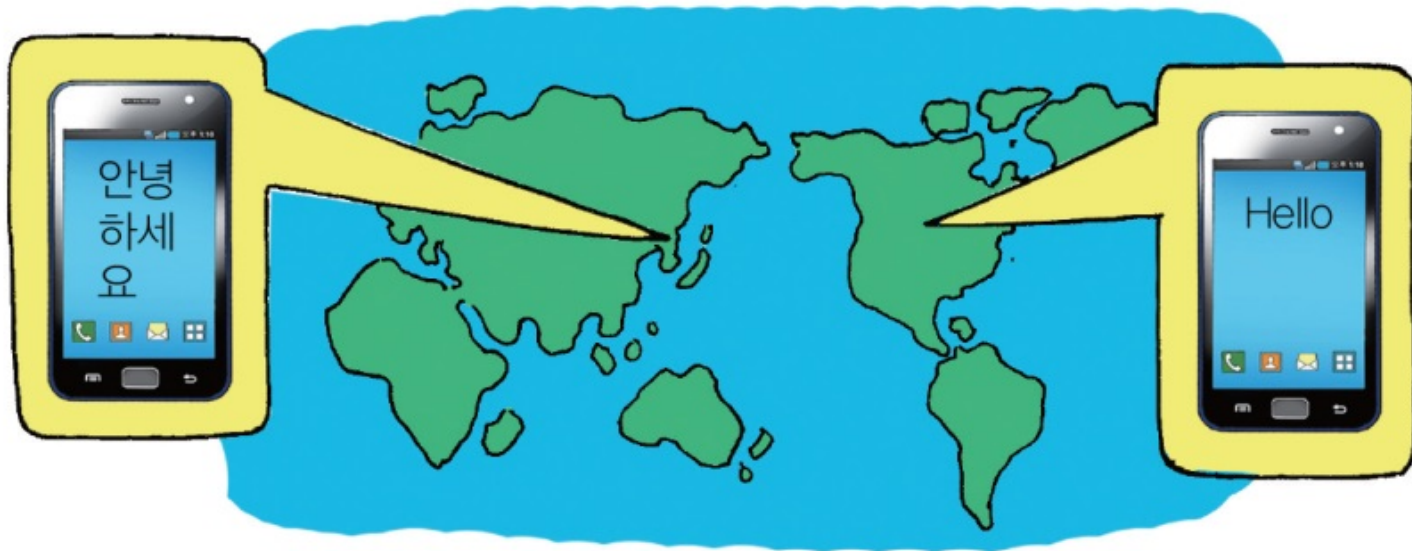
```
drawable/  
drawable-en/  
drawable-en-port/  
drawable-en-notouch-12key/  
drawable-port-ldpi/  
drawable-port-notouch-12key/
```

②

```
drawable-en/  
drawable-en-port/  
drawable-en-notouch-12key/
```

지역화

- 문자열이나 통화, 이미지 같은 여러 가지 리소스들을 사용자가 있는 지역에 따라 변경하는 것



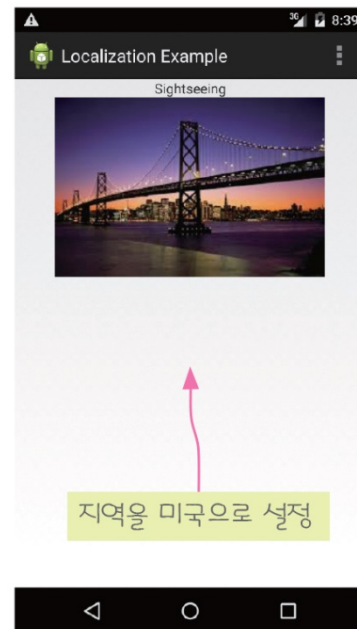
지역화

- 애플리케이션의 사용자 인터페이스의 모든 콘텐츠는 코드와 분리하여 리소스 파일로 보관
- 리소스 준비
 - 기본 리소스(default resource)를 먼저 준비
 - 지역에 따라 사용되는 대체 리소스 추가
- 예)

폴더	내용
res/values/strings.xml	기본 언어(예:영어)
res/values-fr/strings.xml	프랑스어 버전
res/values-ko/strings.xml	한국어 버전

예제

- 설정된 지역에 맞추어서 관광 명소의 이미지를 화면에 표시한다.
- 한국, 미국 만을 지원하도록 작성



레이아웃 파일

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="@string/text"
    />
    <ImageView
        android:id="@+id/ImageView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/image"
    />
</LinearLayout>
```

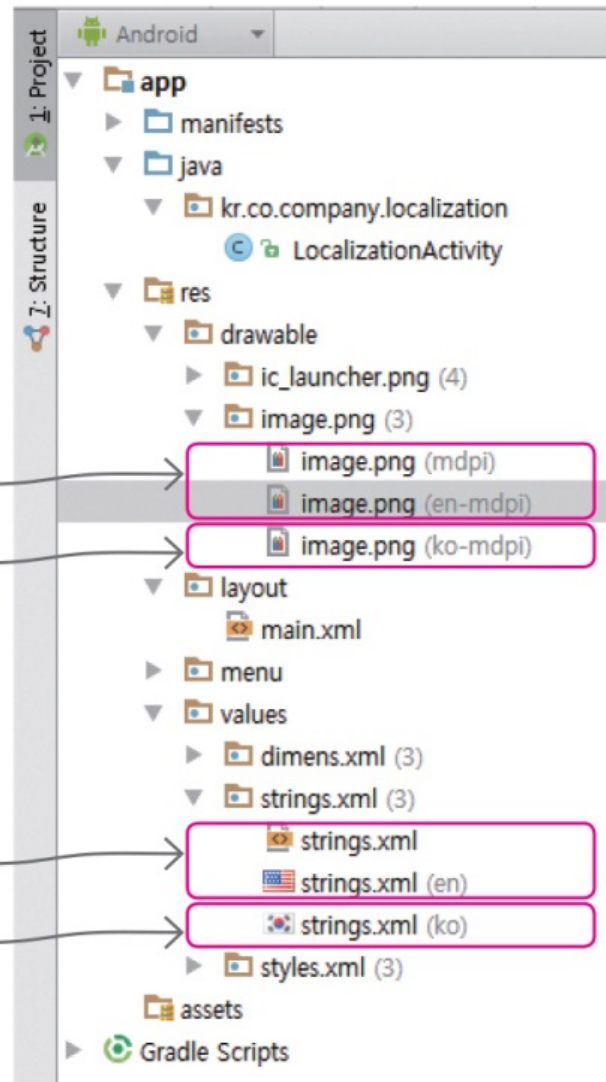
리소스 작성

영어 지역에 사용되는 이미지 리소스

한국 지역에 사용되는 이미지 리소스

영어 지역에 사용되는 텍스트 리소스

한국 지역에 사용되는 텍스트 리소스



XML 파일 작성

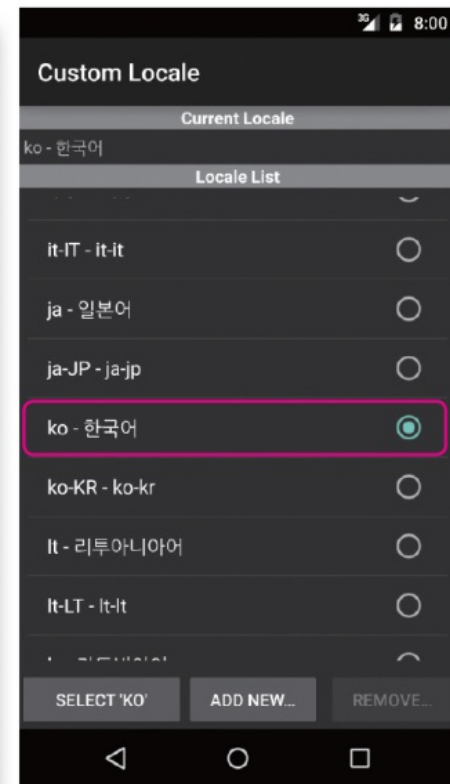
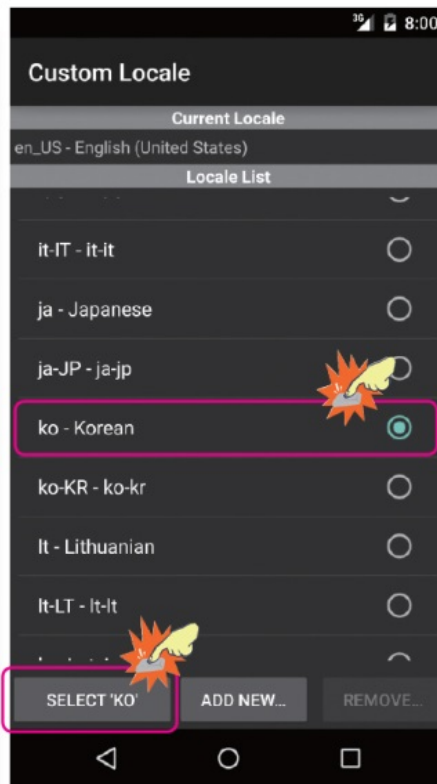
values-ko/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">지역화 예제</string>
    <string name="text">한국의 관광명소</string>
</resources>
```

values-en/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Localization Example</string>
    <string name="text">Sightseeing</string>
</resources>
```

지역 변경 방법



실행 결과



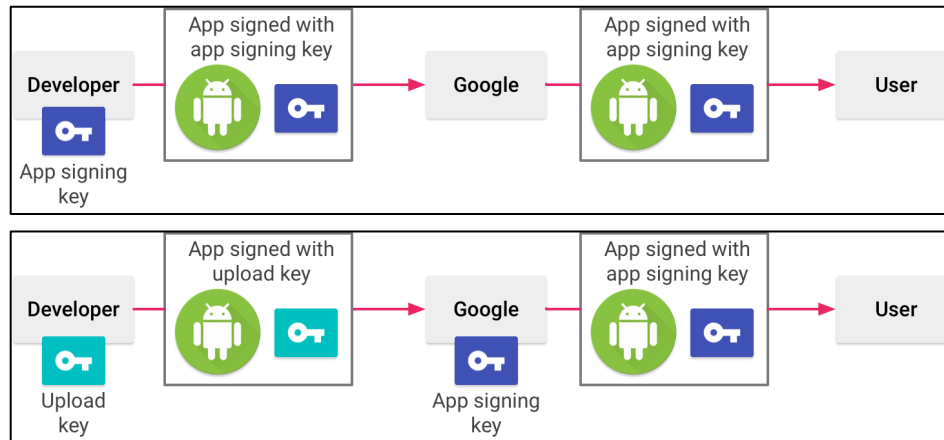
보안

- 안드로이드에서도 애플리케이션이 마음대로 시스템이나 다른 애플리케이션을 건드릴 수 있다면 심각한 위협
- 안드로이드에서 각 애플리케이션은 자신의 프로세스 안에서 실행
- 다른 애플리케이션을 건드릴 수 없다.->
 - Sandboxing
 - One user account per App



애플리케이션 서명(Application Signing)

- 모든 안드로이드 애플리케이션(.apk 파일)은 인증서로 서명
- 인증서는 애플리케이션의 개발자를 식별



- 안드로이드 마켓 배포를 위해서는 필수
 - 인증서
 - JDK의 keytool을 사용하여 keystore 생성, 또는
 - Android Studio의 Build->Generate Signed APK에서 생성
 - apk 생성 시에 keystore 지정

권한 요청하기

- 만약 애플리케이션이 보호된 기능이나 장치 안의 데이터에 접근하려면 AndroidManifest.xml 파일에 필요한 권한을 표시
- 태그 <uses-permission>를 사용

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="kr.co.company.PermissionTest"  
    android:versionCode="1"  
    android:versionName="1.0">
```

...

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

문자 메시지를 받을 수 있는 권한을 요청한다.

권한의 종류

권한	권한 이름	설명
지리 정보 사용	ACCESS_FINE_LOCATION	GPS와 같은 정밀한 위치 정보 사용
전화 걸기	CALL_PHONE	애플리케이션이 전화 걸기 기능 사용
카메라	CAMERA	카메라 사용 가능
일정 정보	READ_CALENDAR	일정 정보 읽기
	WRITE_CALENDAR	일정 정보 쓰기
연락처 정보	READ_CONTACTS	연락처 읽기
	WRITE_CONTACTS	연락처 쓰기
인터넷	INTERNET	인터넷 접속

예제

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.permission"
    android:versionCode="1"
    android:versionName="1.0" >
```

필요한 권한을 요청한다.

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

...

```
</manifest>
```