

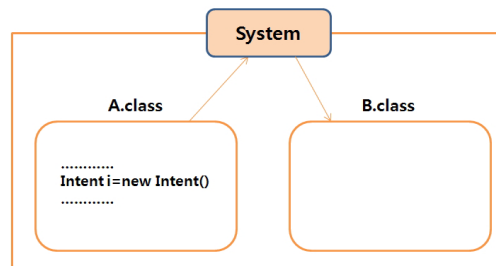


14장. Intent와 구글 기본 앱 연동

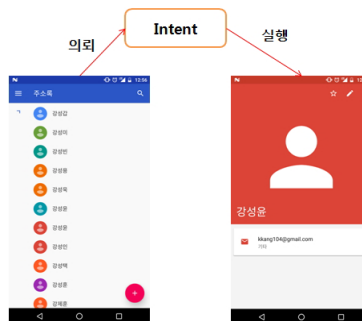
14.1 Intent

14.1.1. 인텐트의 기본 개념

- 안드로이드의 컴포넌트는 독립적으로 실행되어 직접 결합이 발생하지 않는 구조

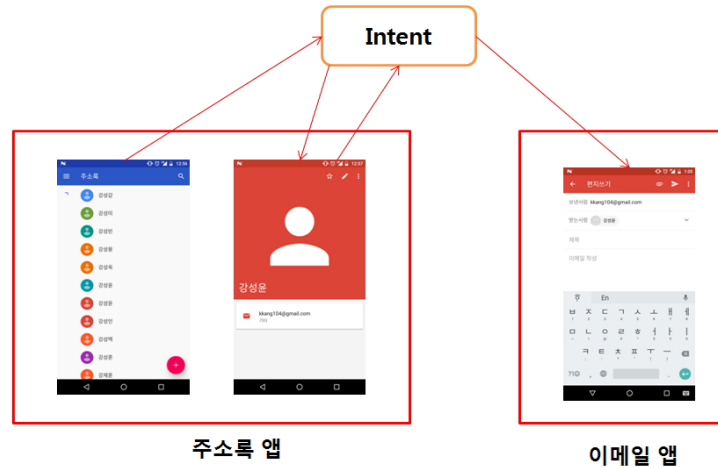


- Intent는 컴포넌트를 실행하기 위해 시스템에 넘기는 정보



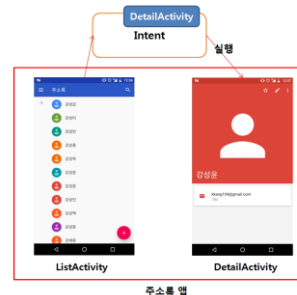
14.1 Intent

- 같은 앱이나 외부 앱의 컴포넌트를 실행



14.1.2. 명시적 인텐트, 암시적 인텐트

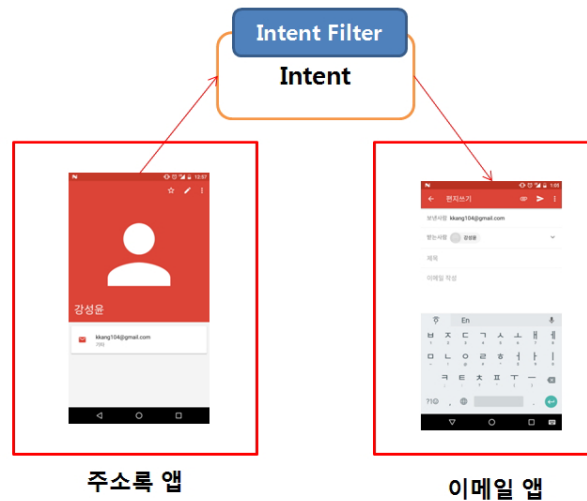
- 명시적 인텐트는 실행하고자 하는 컴포넌트의 클래스명을 인텐트에 담는 방법



14.1 Intent

```
Intent intent=new Intent(this, DetailActivity.class);  
startActivity(intent);
```

- 암시적 인텐트는 클래스명이 아닌 Intent Filter 정보를 활용



```
<activity android:name=".DetailActivity"></activity>
```

- 이 액티비티를 실행하려면 시스템에 등록된 정보와 맞는 정보를 주어야 하는데, 시스템에서 클래스명 정보밖에 안 가지고 있으므로 명시적 인텐트를 이용



14.1 Intent

```
<activity android:name=".DetailActivity">
  <intent-filter>
    <action android:name="com.example.ACTION_VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
  </intent-filter>
</activity>
```

- 클래스명으로 인텐트를 만들어 실행하는 명시적 인텐트도 가능하지만, 클래스명을 명시하지 않고 intent-filter에 등록된 정보를 이용하는 암시적 인텐트도 가능

```
Intent intent=new Intent();
intent.setAction("com.example.com.example.ACTION_VIEW");
startActivity(intent);
```

14.1.3. 인텐트 필터(Intent Filter)

```
<activity android:name=".SomeActivity">
  <intent-filter>
    <action android:name="com.some.ACTION_VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:scheme="http"/>
  </intent-filter>
</activity>
```



14.1 Intent

- action: 컴포넌트가 어떤 능력을 갖추고 있는지에 대한 문자열
- category: 컴포넌트에 대한 추가 정보로 어느 범주의 컴포넌트인지를 표현
- data: data는 컴포넌트를 실행하기 위해 필요한 데이터에 대한 상세 정보

```
<activity android:name=".SomeActivity">  
  <intent-filter>  
    <action android:name="com.some.ACTION_VIEW"/>  
    <category android:name="android.intent.category.DEFAULT"/>  
  </intent-filter>  
</activity>
```

```
Intent intent=new Intent();  
intent.setAction("com.some.ACTION_VIEW");  
startActivity(intent);
```

- startActivity () 함수가 인텐트를 발생시킬 때 자동으로 android.intent.category.DEFAULT 를 추가



14.1 Intent

```
<activity android:name=".SomeActivity">
  <intent-filter>
    <action android:name="com.some.ACTION_VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <category android:name="com.some.category.MYCATEGORY"/>
  </intent-filter>
</activity>
```

- 개발자가 지정한 문자열의 category가 등록되더라도 android.intent.category.DEFAULT는 꼭 선언

```
Intent intent=new Intent();
intent.setAction("com.some.ACTION_VIEW");
startActivity(intent);
```

```
Intent intent=new Intent();
intent.setAction("com.some.ACTION_VIEW");
intent.addCategory("com.some.category.MYCATEGORY");
startActivity(intent);
```



14.1 Intent

```
Intent intent=new Intent();  
intent.setAction("com.some.ACTION_VIEW");  
startActivity(intent);
```

①

action : com.some.ACTION_VIEW
category : android.intent.category.DEFAULT

action : com.some.ACTION_VIEW
category : android.intent.category.DEFAULT
com.some.category.MYCATEGORY

②

```
Intent intent=new Intent();  
intent.setAction("com.some.ACTION_VIEW");  
intent.addCategory("com.some.category.MYCATEGORY");  
startActivity(intent);
```

android.intent.category.DEFAULT
com.some.category.MYCATEGORY



14.1 Intent

- data 선언

```
<activity android:name=".SomeActivity">
  <intent-filter>
    <action android:name="com.some.ACTION_VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:scheme="geo"/>
  </intent-filter>
</activity>
```

- data 정보에는 scheme 이외에 host, port, path가 함께 선언될 수 있다.

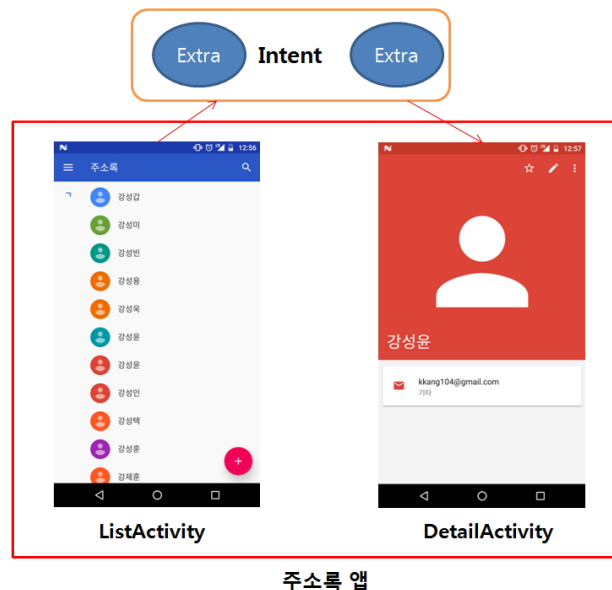
```
Intent intent=new Intent();
intent.setAction("com.some.ACTION_VIEW");
intent.setData(Uri.parse("geo:"));
startActivity(intent);
```

14.1.4. Extra 데이터

- 컴포넌트를 실행하면서 데이터를 전달



14.1 Intent



```
Intent intent=new Intent(this, DetailActivity.class);
intent.putExtra("data1", "hello");
intent.putExtra("data2", 100);
startActivity(intent);
```

- getIntent () 함수를 이용하여 자신을 실행하였던 인텐트 객체를 획득해서 Extra 데이터 추출

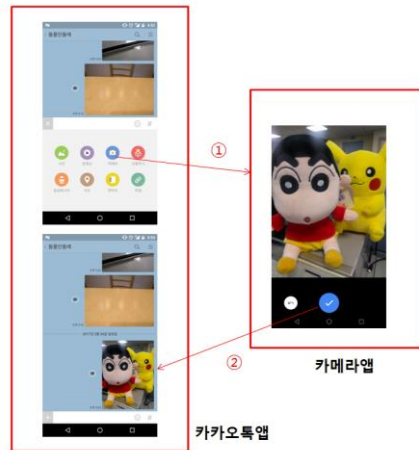
```
Intent intent=getIntent();
String data1=intent.getStringExtra("data1");
int data2=intent.getIntExtra("data2", 0);
```



14.1 Intent

14.1.5. 결과 되돌리기

- 사용자가 뒤로가기 버튼을 누르지 않고도 자바 코드에서 자동으로 화면이 되돌아오게 처리



- `startActivityForResult ()` 함수를 이용

```
Intent intent=new Intent(this, DetailActivity.class);
startActivityForResult(intent, 10);
```

- `startActivityForResult ()` 함수의 두 번째 매개변수는 requestCode 값으로 개발자가 0 이상의 숫자를 지정



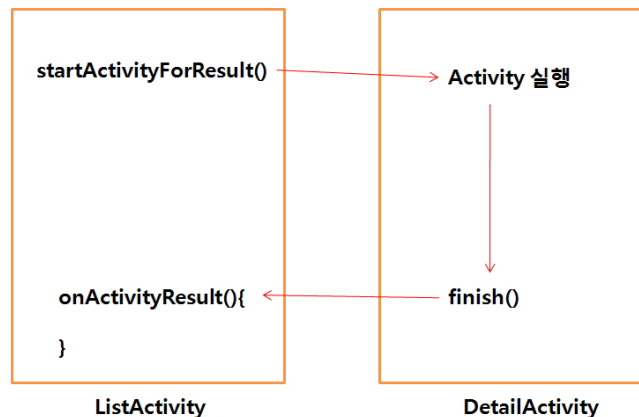
14.1 Intent

- startActivityForResult () 함수로 수행된 액티비티에서 결과를 되돌리고 싶으면 자신의 액티비티를 종료

```
Intent intent=getIntent();
intent.putExtra("location", textView.getText().toString());
setResult(RESULT_OK,intent );
finish();
```

- 액티비티가 종료되면 이전 화면으로 되돌아가며, 이전 액티비티의 onActivityResult () 함수가 자동으로 호출

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode==10 && resultCode==RESULT_OK){
        //...    }
    }
```

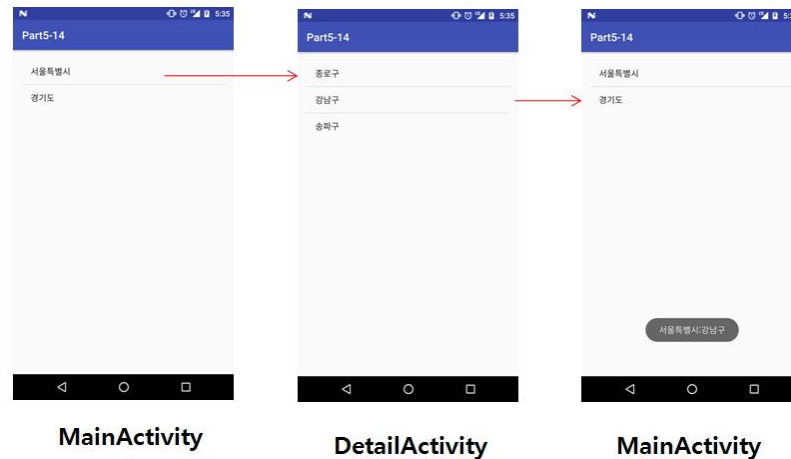


Step by Step 14-1 - 인턴트

Intent를 이용하는 실습

- 첫번째 Activity에서 항목을 선택했을때 두번째 Activity에서 선택한 항목의 서브 항목을 출력시키고 서브 항목을 선택시 자동으로 화면을 첫번째 화면으로 되돌리는 테스트

1. Module 생성
2. DetailActivity 추가
3. 파일 복사
4. MainActivity 작성
5. DetailActivity 작성
6. 실행



14.2 구글 기본 앱 연동

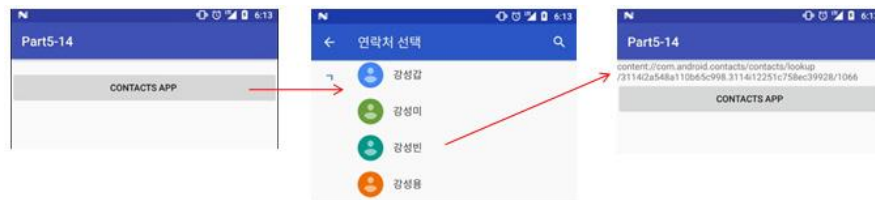
14.2.1. 주소록 앱

- 주소록 앱을 연동하여 주소록 목록 화면을 띄우기

```
Intent intent = new Intent(Intent.ACTION_PICK);  
intent.setData(ContactsContract.Contacts.CONTENT_URI);  
startActivityForResult(intent, 10);
```

- 결과를 되돌려받기

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if(requestCode==10 && resultCode==RESULT_OK){  
        String result=data.getDataString();  
        resultView.setText(result);  
    }  
}
```



- 인텐트의 data.getDataString () 함수를 이용하여 넘어온 URL 값 획득
- URL 정보의 맨 마지막 단어가 선택한 홍길동의 식별자



14.2 구글 기본 앱 연동

인텐트를 발생시킬 때 setData () 함수의 Uri 값을 다양화 해서 목록 화면에 출력되는 데이터를 조정

- ContactsContract.Contacts.CONTENT_URI: 사람 이름
- ContactsContract.CommonDataKinds.Phone.CONTENT_URI: 사람 이름과 전화번호
- ContactsContract.CommonDataKinds.Email.CONTENT_URI: 사람 이름과 이메일
- 한 사람에 대한 상세보기 화면

```
Intent intent = new Intent(Intent.ACTION_VIEW);  
intent.setData(Uri.parse(ContactsContract.Contacts.CONTENT_URI+"/"+1066));  
startActivity(intent);
```

14.2.2. 카메라 앱

- 인텐트 발생만으로도 쉽게 사진 촬영과 촬영된 사진 데이터 획득
- 섬네일로 결과 받기
- 이 방식은 카메라 앱으로 촬영한 사진이 파일로 저장되지 않으며 사진 데이터 자체가 전달되는 구조

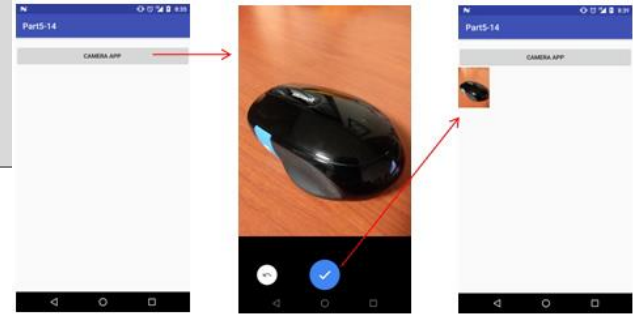
```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
startActivityForResult(intent, 30);
```



14.2 구글 기본 앱 연동

- 결과 획득

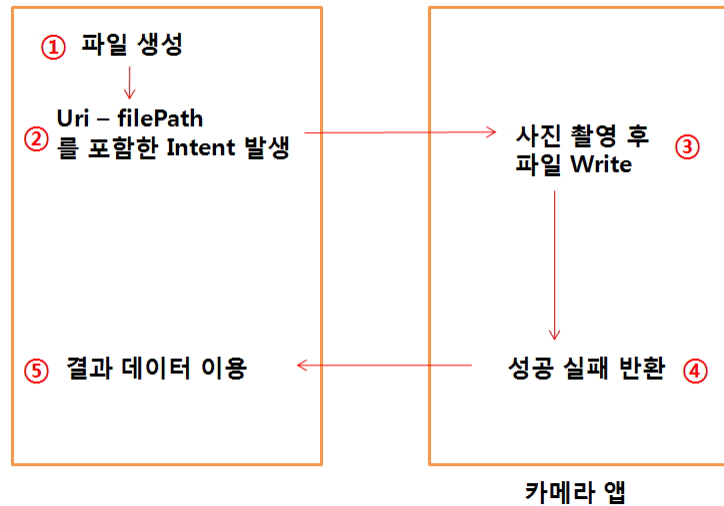
```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if(requestCode==30 && resultCode==RESULT_OK){  
        Bitmap bitmap=(Bitmap)data.getExtras().get("data");  
        resultImageView.setImageBitmap(bitmap);  
    }  
}
```



- 파일 공유 방법
- 크기를 줄이지 않고 카메라로 촬영한 크기 그대로 이용
- 임의의 경로에 파일을 하나 만들고, 이 파일 경로를 카메라 앱에 전달하여 카메라 앱에서 사진 촬영 데이터를 파일에 쓴(Write) 다음, 성공 여부를 반환



14.2 구글 기본 앱 연동



- 파일 준비

```
String dirPath = Environment.getExternalStorageDirectory()
    .getAbsolutePath() + "/myApp";
File dir = new File(dirPath);
if (!dir.exists()) {
    dir.mkdir();
}

filePath = File.createTempFile("IMG", ".jpg", dir);
if (!filePath.exists()) {
    filePath.createNewFile();
}
```



14.2 구글 기본 앱 연동

- 외부 앱에 파일 정보 공유 방법
- API Level 24 이전 버전

```
intent.putExtra( MediaStore.EXTRA_OUTPUT, Uri.fromFile(filePath) );
```

- API Level 24 버전
- 앱 외부에 file:// URI의 노출을 금지하는 스트릭트(Strict) 모드 정책을 적용
- content:// URI를 보내고 이 URI에 대해 임시 액세스 권한을 부여
- FileProvider 클래스를 이용
- FileProvider를 이용하려면 res / xml 폴더에 임의의 이름으로 XML 파일 필요

```
<paths xmlns:android="http://schemas.android.com/apk/res/android">  
  <external-path name="myfiles" path="."/>  
</paths>
```



14.2 구글 기본 앱 연동

- AndroidManifest.xml에서 FileProvider를 등록하면서 설정

```
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="com.example.test5_14.provider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/provider_paths"/>
</provider>
```

- 등록한 FileProvider를 자바 코드에서 이용해 공유하려는 파일 정보의 Uri 값 획득

```
Uri photoURI = FileProvider.getUriForFile(Lab2Activity.this,
    BuildConfig.APPLICATION_ID + ".provider", filePath);
```

- Uri 값을 카메라 앱 실행을 위한 인텐트의 Extra 데이터로 설정

```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
intent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
startActivityForResult(intent, 40);
```



14.2 구글 기본 앱 연동

- 이미지 로딩으로 인한 OutOfMemoryException 문제
- 이미지의 크기를 줄여서 로딩

```
Bitmap bitmap = BitmapFactory.decodeFile(filePath.getAbsolutePath())
```

Bitmap은 이미지를 표현하는 클래스로 BitmapFactory 클래스로 생성

- BitmapFactory.decodeByteArray(): byte[] 배열로 Bitmap 생성
 - BitmapFactory.decodeFile(): 파일 경로를 주면 FileInputStream을 만들어서 decodeStream 이용
 - BitmapFactory.decodeResource(): Resource 폴더에 저장된 파일
 - BitmapFactory.decodeStream(): InputStream으로 Bitmap 생성
-
- Options 클래스의 inSampleSize

```
BitmapFactory.Options imgOptions = new BitmapFactory.Options();  
imgOptions.inSampleSize = 10;  
Bitmap bitmap = BitmapFactory.decodeFile(filePath.getAbsolutePath(), imgOptions);
```



14.2 구글 기본 앱 연동

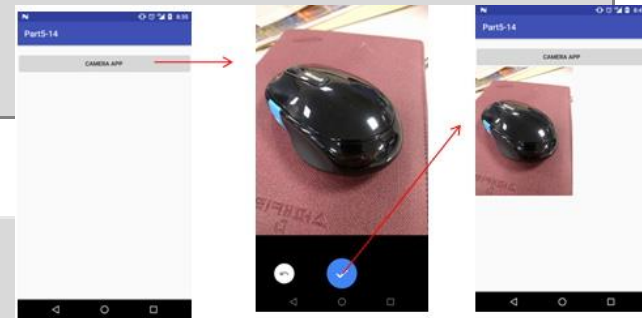
- 비율을 동적으로 결정
- 이미지의 가로세로 크기 획득

```
BitmapFactory.Options options = new BitmapFactory.Options();
options.inJustDecodeBounds = true;
try {
    InputStream in = new FileInputStream(filePath);
    BitmapFactory.decodeStream(in, null, options);
    in.close();
    in = null;
} catch (Exception e) {
    e.printStackTrace();
}
```

```
final int height = options.outHeight;
final int width = options.outWidth;
```

- 이미지 크기를 비교해서 적절한 inSampleSize를 계산

```
if (height > reqHeight || width > reqWidth) {
    final int heightRatio = Math.round((float) height / (float) reqHeight);
    final int widthRatio = Math.round((float) width / (float) reqWidth);
    inSampleSize = heightRatio < widthRatio ? heightRatio : widthRatio;
}
```



14.2 구글 기본 앱 연동

- 동영상 촬영

```
Intent intent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
intent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, 0);
intent.putExtra(MediaStore.EXTRA_DURATION_LIMIT, 20);
intent.putExtra(MediaStore.EXTRA_SIZE_LIMIT, 1024 * 1024 * 10);
intent.putExtra(MediaStore.EXTRA_OUTPUT, videoURI);
startActivityForResult(intent, 20);
```

14.2.3. 갤러리 앱

- 갤러리 앱의 목록 액티비티

```
Intent intent = new Intent(Intent.ACTION_PICK);
intent.setType(MediaStore.Images.Media.CONTENT_TYPE);
intent.setData(MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
startActivityForResult(intent, 20);
```

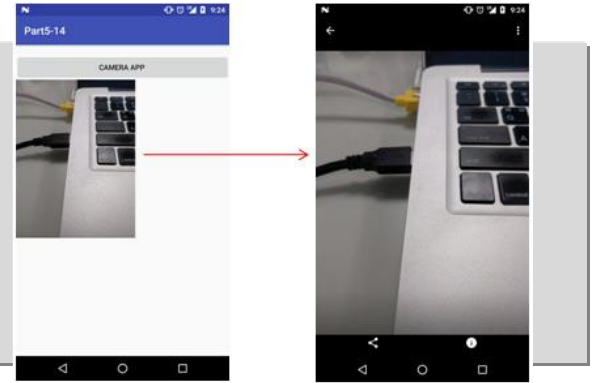
```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode==20 && resultCode==RESULT_OK){
        String result=data.getDataString();
        resultView.setText(result);
    }
}
```



14.2 구글 기본 앱 연동

- 사진 이미지를 사용자에게 전체화면으로 보여 주기

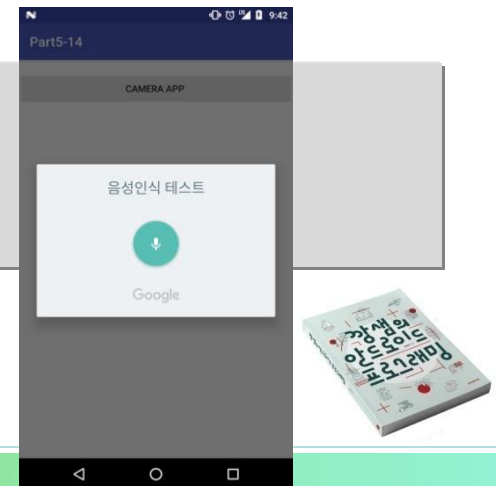
```
Intent intent = new Intent();  
intent.setAction(Intent.ACTION_VIEW);  
Uri photoURI = FileProvider.getUriForFile(Lab2Activity.this,  
    BuildConfig.APPLICATION_ID + ".provider",filePath);  
intent.setDataAndType(photoURI,"image/*");  
//외부 앱에서 우리 데이터를 이용하는 것이므로  
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);  
startActivity(intent);
```



- 파일 정보를 갤러리 앱에 전달하기 위해 FileProvider를 이용
- intent. addFlags (Intent.FLAG_GRANT_READ_URI_PERMISSION); 코드로 인텐트에 정보를 설정

14.2.4. 음성인식 앱

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,  
    RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);  
intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "음성인식 테스트");  
startActivityForResult(intent, 50);
```



14.2 구글 기본 앱 연동

- 음성인식에 의한 결과 문자열

```
ArrayList<String> results = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);  
String result=results.get(0);
```

14.2.5. 기타 앱 연동

- 지도 앱 연동

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("geo:37.5662952,126.9779451"));  
startActivity(intent);
```



- 브라우저 앱

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.seoul.go.kr"));  
startActivity(intent);
```

- 전화 앱

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

```
Intent intent=new Intent(Intent.ACTION_CALL, Uri.parse("tel:02-120"));  
startActivity(intent);
```



Step by Step 14-2 – 구글 기본 앱 연동

구글의 기본앱 연동을 테스트

•화면은 버튼으로 단순 처리해서 버튼 클릭시 각 앱을 연동시키는 방식으로 테스트

1. Activity 생성
2. 파일 복사
3. demens.xml 파일 작성
4. AndroidManifest.xml 작성
5. Lab14_2Activity 작성
6. Lab14_2Activity.java 실행

