

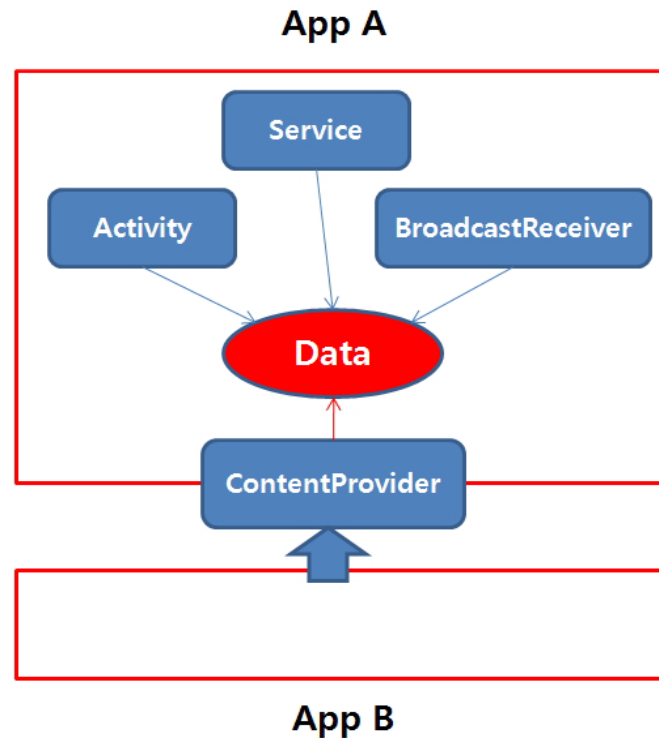
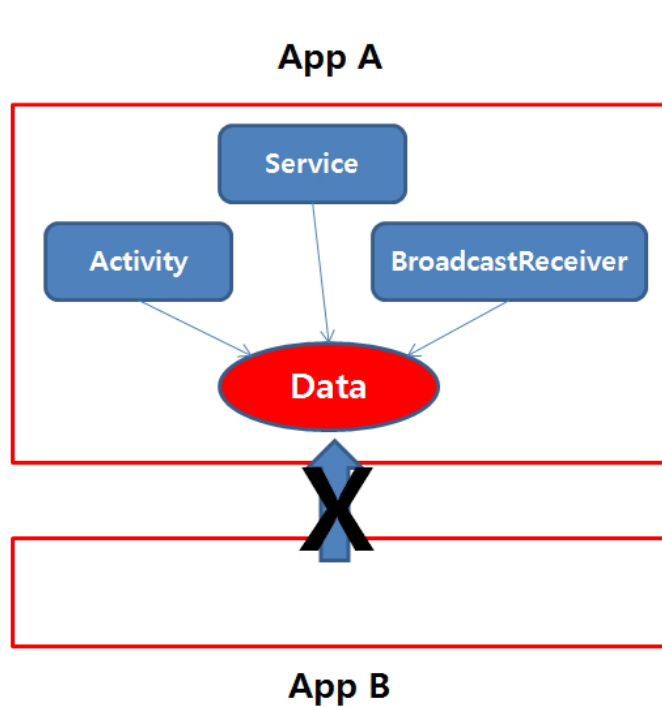


21장. 콘텐츠 프로바이더

21.1 ContentProvider이해

21.1.1. 콘텐츠 프로바이더 구조

- 콘텐츠 프로바이더는 앱 간의 데이터 공유를 목적으로 사용 되는 컴포넌트



21.1 ContentProvider이해

21.1.2. 콘텐츠 프로바이더 작성법

```
public class MyContentProvider extends ContentProvider {
    public MyContentProvider() { }
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        return 0;
    }
    public String getType(Uri uri) {
        throw new UnsupportedOperationException("Not yet implemented");
    }
    public Uri insert(Uri uri, ContentValues values) {
        return null;
    }
    public boolean onCreate() {
        return false;
    }
    public Cursor query(Uri uri, String[] projection, String selection,
        String[] selectionArgs, String sortOrder) {
        return null;
    }
    public int update(Uri uri, ContentValues values, String selection,
        String[] selectionArgs) {
        return 0;
    }
}
```

21.1 ContentProvider이해

- AndroidManifest.xml에 등록

```
<provider  
    android:name=".MyContentProvider"  
    android:authorities="com.example.test.Provider"  
    android:enabled="true"  
    android:exported="true"></provider>
```

- URL은 규칙
- content:// com.example.test.Provider
- content:// com.example.test.Provider/user
- content:// com.example.test.Provider/user/1

content:// com.example.test.Provider/user/1

Scheme host path

21.1 ContentProvider 이해

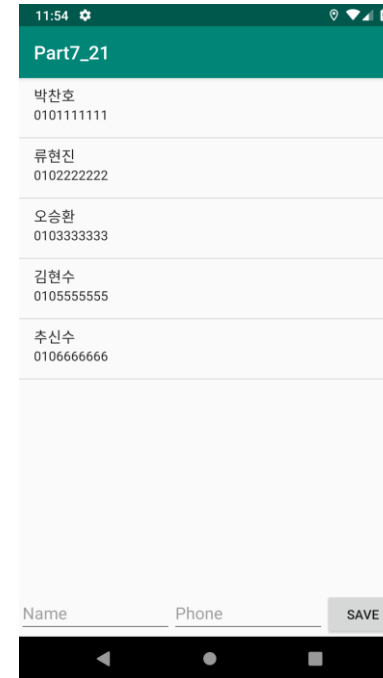
21.1.3. 콘텐츠 프로바이더 이용

```
Cursor cursor=getContentResolver().query(uri, null, null, null, null);
```

- query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)
- insert(Uri url, ContentValues values)
- update(Uri uri, ContentValues values, String where, String[] selectionArgs)
- delete(Uri url, String where, String[] selectionArgs)

Step by Step 21-1 - ContentProvider

- 모듈 생성
- 파일 복사
- Part7_21_provider 모듈 작업
- Part7_21 모듈 작업



21.2 구글 기본 앱의 ContentProvider 이용

21.2.1. 주소록 앱 데이터 획득

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

- 주소록의 목록 액티비티

```
Intent intent = new Intent(Intent.ACTION_PICK);  
intent.setData(Uri.parse("content://com.android.contacts/data/phones"));  
startActivityForResult(intent, 10);
```

- 콘텐츠 프로바이더가 이용

```
Cursor cursor = getContentResolver().query(  
    ContactsContract.Data.CONTENT_URI,  
    new String[] {  
        ContactsContract.Contacts.DISPLAY_NAME,  
        ContactsContract.CommonDataKinds.Phone.NUMBER },  
    ContactsContract.Data._ID + "=" + id, null, null);
```


21.2 구글 기본 앱의 ContentProvider 이용

21.2.2. 갤러리 앱 데이터 획득

- API Level 16 하위 버전

```
if (Build.VERSION.SDK_INT < Build.VERSION_CODES.JELLY_BEAN) {  
    //jelly bean 하위에서는 하나만 선택하게..  
    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);  
}
```

- onActivityResult () 함수

```
String[] projection = {MediaStore.Images.Media.DATA};  
Cursor cursor = getContentResolver().query(data.getData(), projection, null, null, null);  
cursor.moveToFirst();  
String filePath = cursor.getString(0);
```


21.2 구글 기본 앱의 ContentProvider 이용

- API Level 16 이상 버전
- 인텐트를 발생

```
Intent intent = new Intent();
intent.setType("image/*");
intent.putExtra(Intent.EXTRA_ALLOW_MULTIPLE, true);
intent.setAction(Intent.ACTION_GET_CONTENT);
startActivityForResult(intent, 30);
```

- 결과 획득

```
if (data.getClipData() != null) {
    ClipData clipData = data.getClipData();
    for (int i = 0; i < clipData.getItemCount(); i++) {
        ClipData.Item item = clipData.getItemAt(i);
        Uri uri = item.getUri();
        if ("com.android.providers.media.documents".equals(uri.getAuthority()) &&
            Build.VERSION.SDK_INT >= 19) {
            //...
        } else if ("external".equals(uri.getPathSegments().get(0))) {
            //...
        }
    }
} else {
}
```

21.2 구글 기본 앱의 ContentProvider 이용

- ClipData 타입으로 전달되었을 때 API Level 19(Android 4.4)부터 결과 데이터의 Uri 표현식이 이전 버전과 달라져서 이 또한 구분해서 처리
- ① 도큐먼트식 경로:
content://com.android.providers.media.documents/document/image:3A35260
- ② 세그먼트식 경로:
content://media/external/images/media/....
- 세그먼트식 이미지 파일 경로 획득

```
String selection = MediaStore.Images.Media._ID + "=?";
String[] selectionArgs = new String[] {uri.getLastPathSegment()};

String column = "_data";
String[] projection = {column};

Cursor cursor = getContentResolver().query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, projection, selection,
selectionArgs, null);
String filePath=null;
if (cursor != null && cursor.moveToFirst()) {
    int column_index = cursor.getColumnIndexOrThrow(column);
    filePath = cursor.getString(column_index);
}
cursor.close();
```

21.2 구글 기본 앱의 ContentProvider 이용

- 도큐먼트식 이미지 파일 경로 획득

```
String docId = DocumentsContract.getDocumentId(uri);
String[] split = docId.split(":");
String type = split[0];
Uri contentUri = null;
if ("image".equals(type)) { //video, audio type 도 가능
    contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
}

String selection = MediaStore.Images.Media._ID + "=?";
String[] selectionArgs = new String[]{split[1]};

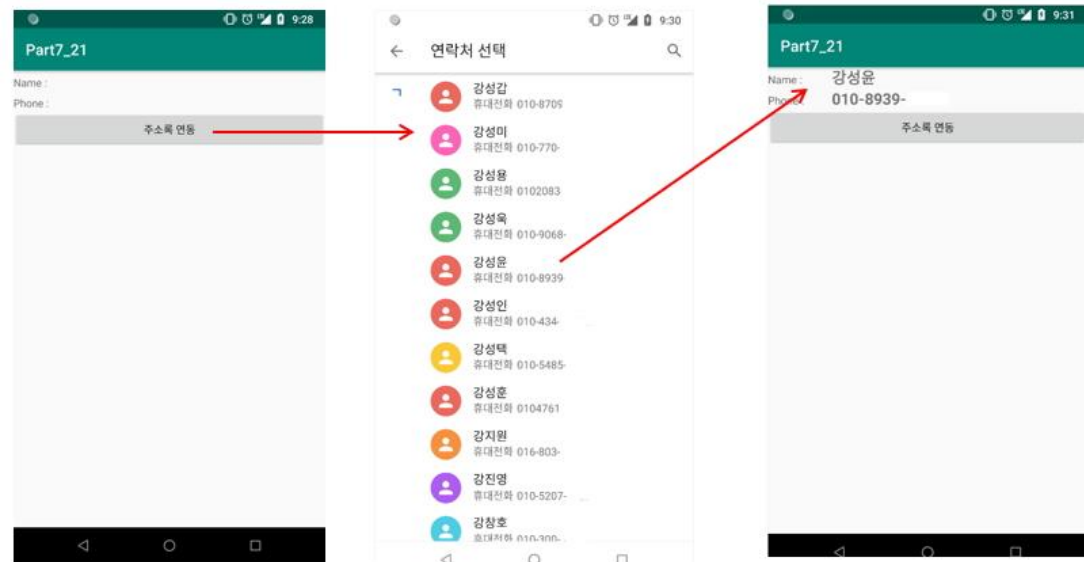
String column = "_data";
String[] projection = {column};

Cursor cursor = context.getContentResolver().query(contentUri, projection, selection, selectionArgs, null);

String filePath = null;
if (cursor != null && cursor.moveToFirst()) {
    int column_index = cursor.getColumnIndexOrThrow(column);
    filePath = cursor.getString(column_index);
}
cursor.close();
```

Step by Step 21-2 – 주소록, 갤러리 앱 연동

- 액티비티 생성
- 파일 복사
- AndroidManifest.xml 작업
- Lab21_2Activity 작성
- 실행



21.3 갤러리 앱 연동과 이미지 이용을 위한 라이브러리

21.3.1 Glide 라이브러리

2014년 구글 IO 행사에서 발표

Glide를 이용하여 HTTP 연동에 의한 서버 이미지뿐 아니라, 리소스, 파일 이미지도 쉽게 가져올 수 있으며, 섬네일 보기, 로딩 이미지 출력, 동영상 스틸 보기 등의 기능을 지원

```
implementation 'com.github.bumptech.glide:glide:3.8.0'
```

리소스 이미지

```
Glide.with(this)
    .load(R.drawable.photos)
    .into(resourceView);
```

파일 이미지

```
File file = new File(path);
Uri imageUri = Uri.fromFile(file);
Glide.with(this)
    .load(imageUri)
    .into(imageView);
```

21.3 갤러리 앱 연동과 이미지 이용을 위한 라이브러리

네트워크 이미지

```
Glide.with(this)
    .load(url)
    .into(imageView);
```

GIF 이미지

```
Glide.with(this)
    .load(R.raw.loading)
    .asGif()
    .diskCacheStrategy(DiskCacheStrategy.SOURCE)
    .into(imageView);
```

특정 크기로 로딩되길 원한다면 `override ()` 함수.
`placeholder ()` 함수를 이용하여 미리 보이는 이미지를 지정가능.
이미지 로딩에 실패할 때 보여질 이미지도 `error ()` 함수로 지정 가능

```
Glide.with(this)
    .load(url)
    .override(400, 400)
    .placeholder(R.raw.loading)
    .error(R.drawable.error)
    .into(imageView);
```

21.3 갤러리 앱 연동과 이미지 이용을 위한 라이브러리

획득한 이미지를 BitmapImageViewTarget의 생성자에 설정한 ImageView에 출력

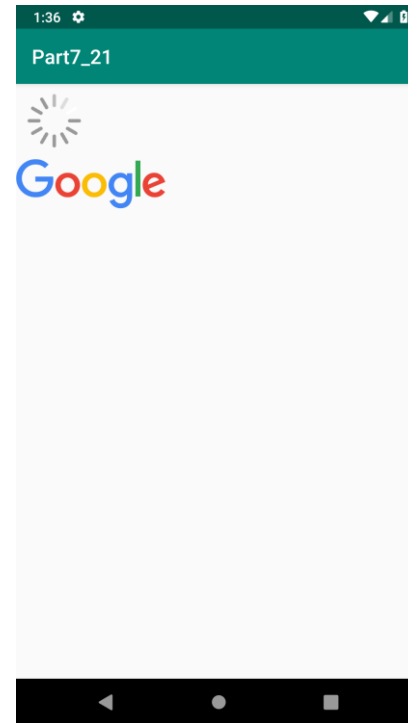
```
Glide.with(this)
    .load(imageUri)
    .asBitmap()
    .into(new BitmapImageViewTarget(imageView){
        @Override
        protected void setResource(Bitmap resource) {
            super.setResource(resource);
        }
    });
```

이미지 데이터 획득

```
Glide.with(this)
    .load(imageUri)
    .asBitmap()
    .into(new SimpleTarget<Bitmap>(250, 250) {
        @Override
        public void onResourceReady(Bitmap resource, GlideAnimation<? super
        Bitmap> glideAnimation) {
            //...
        }
    });
```


Step by Step 21-3 – Glide API

- 액티비티 생성
- 파일 복사
- 그레이들 설정
- Lab21_3Activity 작성
- 실행



21.4 Matisse 라이브러리

Matisse (<https://github.com/zhihu/Matisse>)는 갤러리를 연동하기 위한 라이브러리

```
implementation 'com.zhihu.android:matisse:0.5.2-beta4'
```

이미지 목록 화면

```
Matisse.from(this)
    .choose(MimeType.ofAll())
    .countable(true)
    .maxSelectable(9)
    .spanCount(3)
    .imageEngine(new GlideEngine())
    .forResult(100);
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 100 && resultCode == RESULT_OK) {
        List<Uri> mSelected = Matisse.obtainResult(data);
    }
}
```

Step by Step 21-4 – Matisse 라이브러리

- 액티비티 생성
- 파일 복사
- 그레이들 설정
- AndroidManifest.xml 작업
- Lab21_4Activity 작성
- 실행

