
App 포트폴리오

취미DB를 활용한 커스텀 리스트

이 름	김민성
제작 기간	2016/12/14 ~ 2016/12/28

1

개요

MySQL을 이용하여, 영화 관련 DB를 구축한뒤, 안드로이드 앱에서 JSON을 활용하여, 서버에 접속한뒤, DB의 내용을 받아와 커스텀 리스트형태로, 출력하고, 리스트 아이템을 클릭하면 ,영화의 줄거리 및 영화의 세부사항들을 보여주도록 프로그래밍 하였습니다.

2

소스코드

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="s20120209.ce.kumoh.ac.kr.testlist">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".MovieActivity"/>
    </application>

</manifest>
```

MainActivity.java

```
package s20120209.ce.kumoh.ac.kr.testlist;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
```

```

import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Cache;
import com.android.volley.Network;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.BasicNetwork;
import com.android.volley.toolbox.DiskBasedCache;
import com.android.volley.toolbox.HurlStack;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.NetworkImageView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity
    implements AdapterView.OnItemClickListener {

    protected ArrayList<MovieInfo> mArray = new ArrayList<MovieInfo>();

    public static final String MOVIE_TAG = "GundamTag";

    protected JSONObject mResult = null;

    protected ListView mList;
    protected MovieAdapter mAdapter;
    protected RequestQueue mQueue = null; //인터넷에 뭔가 요청할때 사용하는것은 이 mQueue에 넣어줌.
    protected ImageLoader mImageLoader = null; //네트워크 이미지를 쓸때 사용하는 변수

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

```

```

        mAdapter = new MovieAdapter(this,
            R.layout.listitem, mArray);

        mList = (ListView) findViewById(R.id.list);
        mList.setAdapter(mAdapter);
        mList.setOnItemClickListener(this); //리스너 등록

        //Volley를위한 설정
        Cache cache = new DiskBasedCache(getCacheDir(), 1024 * 1024);
        Network network = new BasicNetwork(new HurlStack());
        mQueue = new RequestQueue(cache, network);
        mQueue.start();

        mImageLoader = new ImageLoader(mQueue,
            new LruBitmapCache(LruBitmapCache.getCacheSize(this)));

        requestMovie();

    }

    protected void requestMovie() {
        String url = "http://192.168.0.2/customlist/selectjson.php";

        JsonObjectRequest jsonObjRequest = new JsonObjectRequest(Request.Method.GET, url, null,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    mResult = response;
                    drawList(); //잘 응답되었다면, drawList() 호출
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(MainActivity.this, "서버 에러", Toast.LENGTH_LONG).show();
                }
            }
        );

        jsonObjRequest.setTag(MOVIETAG); //후에 Queue에 남아있는 데이터 삭제를 위해서 사용
        mQueue.add(jsonObjRequest); //Queue에 데이터 삽입
    }

    public void drawList() {
        mArray.clear();
        try {

```

```

JSONArray jsonMainNode = mResult.getJSONArray("list");

for (int i = 0; i < jsonMainNode.length(); i++) {
    JSONObject jsonChildNode = jsonMainNode.getJSONObject(i);

    String title = jsonChildNode.getString("title");
    Log.i("title", title);

    String director = jsonChildNode.getString("director");
    Log.i("director", director);

    String playdate = jsonChildNode.getString("playdate");
    Log.i("playdate", playdate);

    String genre = jsonChildNode.getString("genre");
    Log.i("genre", genre);

    String logo = jsonChildNode.getString("logo");
    Log.i("logo", logo);

    String grade = jsonChildNode.getString("grade");
    Log.i("grade", grade);

    String content = jsonChildNode.getString("content");
    Log.i("content", content);

    String poster = jsonChildNode.getString("poster");
    Log.i("poster", poster);

    mArray.add(new MovieInfo(title, director,playdate,genre,grade,content, poster,logo));
}
} catch (JSONException | NullPointerException e) {
    Toast.makeText(getApplicationContext(),
        "Error" + e.toString(), Toast.LENGTH_LONG).show();
    mResult = null;
}

mAdapter.notifyDataSetChanged();
}

@Override
protected void onStop() {
    super.onStop();
    if (mQueue != null)
        mQueue.cancelAll(MOVIETAG);
}

@Override

```

```

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

    Intent intent = new Intent(this, MovieActivity.class);

    String [] data = new String[8];
    data[0] = mArray.get(i).getTitle();
    data[1] = mArray.get(i).getDirector();
    data[2] = mArray.get(i).getPlaydate();
    data[3] = mArray.get(i).getGenre();
    data[4] = mArray.get(i).getGrade();
    data[5] = mArray.get(i).getContent();
    data[6] = mArray.get(i).getPoster();
    data[7] = mArray.get(i).getLogo();
    intent.putExtra("data", data);
    startActivity(intent);
}

public class MovieInfo {
    String title;        //영화 제목
    String director;     //영화 감독
    String playdate;     //영화 개봉일
    String genre;        //영화 장르
    String grade;        //영화 등급
    String content;      //영화 줄거리
    String poster;       //영화 포스터 (이미지)
    String logo;         //영화사 로고 (이미지)

    public MovieInfo(String title, String director, String playdate, String genre, String grade,
String content, String poster, String logo) {
        this.title = title;
        this.director = director;
        this.playdate = playdate;
        this.genre = genre;
        this.grade = grade;
        this.content = content;
        this.poster = poster;
        this.logo = logo;
    }

    public String getTitle() {
        return title;
    }

    public String getDirector() {
        return director;
    }

    public String getPlaydate() {

```

```

        return playdate;
    }

    public String getGenre() {
        return genre;
    }

    public String getGrade() {
        return grade;
    }

    public String getContent() {
        return content;
    }

    public String getPoster() {
        return poster;
    }

    public String getLogo() {
        return logo;
    }
}

static class MovieViewHolder {
    TextView txTitle;
    TextView txDirector;
    TextView txPlaydate;
    NetworkImageView imPoster;
    NetworkImageView imLogo;
}

public class MovieAdapter extends ArrayAdapter<MovieInfo> {
    MovieViewHolder holder;

    public MovieAdapter(Context context, int resource, List<MovieInfo> objects) {
        super(context, resource, objects);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        if (convertView == null) { //처음 아이템 생성
            convertView = getLayoutInflater().inflate(R.layout.listitem, parent, false);

            holder = new MovieViewHolder();
            holder.txTitle = (TextView) convertView.findViewById(R.id.title);
            holder.txDirector = (TextView) convertView.findViewById(R.id.director);
            holder.txPlaydate = (TextView) convertView.findViewById(R.id.playdate);

```

```

        holder.imPoster = (NetworkImageView) convertView.findViewById(R.id.poster);
        convertView.setTag(holder);
        holder.imLogo = (NetworkImageView) convertView.findViewById(R.id.logo);
        convertView.setTag(holder);

    } else { //재활용
        holder = (MovieViewHolder) convertView.getTag();

    }
    holder.txTitle.setText(getItem(position).getTitle());
    holder.txDirector.setText(getItem(position).getDirector());
    holder.txPlaydate.setText(getItem(position).getPlaydate());
    holder.imPoster.setImageUrl("http://192.168.0.2/customlist/"
        + getItem(position).getPoster(), mImageLoader);
    holder.imLogo.setImageUrl("http://192.168.0.2/customlist/"
        + getItem(position).getLogo(), mImageLoader);
    return convertView;
}
}
}
}

```

content_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/content_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="s20120209.ce.kumoh.ac.kr.testlist.MainActivity"
    tools:showIn="@layout/activity_main">

    <ListView
        android:id="@+id/list"

```



```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Hello World!" />
</LinearLayout>
```

listitem.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <com.android.volley.toolbox.NetworkImageView
        android:id="@+id/poster"
        android:layout_width="80dp"
        android:layout_height="100dp"
        android:layout_alignParentLeft="true"
        android:padding="5dp"
        android:scaleType="centerCrop" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="3dp"
        android:orientation="vertical"
        android:padding="5dp">

        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#000"
            android:textSize="20sp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:padding="5dp">

            <com.android.volley.toolbox.NetworkImageView
                android:id="@+id/logo"
                android:layout_width="50dp"
                android:layout_height="60dp"
                android:padding="5dp"
                android:scaleType="centerCrop" />
```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="2dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="감독 : "
            android:textColor="#00f"
            android:textSize="15sp" />

        <TextView
            android:id="@+id/director"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#00f"
            android:textSize="15sp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="2dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="개봉일 : "
            android:textColor="#ff0000"
            android:textSize="15sp" />

        <TextView
            android:id="@+id/playdate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#ff0000"
            android:textSize="15sp" />

```

```

        </LinearLayout>

        </LinearLayout>

    </LinearLayout>
</LinearLayout>
</LinearLayout>

```

MovieActivity.java

```

package s20120209.ce.kumoh.ac.kr.testlist;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

import com.android.volley.Cache;
import com.android.volley.Network;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.BasicNetwork;
import com.android.volley.toolbox.DiskBasedCache;
import com.android.volley.toolbox.HurlStack;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.NetworkImageView;

/**
 * Created by Administrator on 2016-12-05.
 */

public class MovieActivity extends AppCompatActivity {

    String[] data;
    String title;        //영화 제목
    String director;     //영화 감독
    String playdate;     //영화 개봉일
    String genre;        //영화 장르
    String grade;        //영화 등급
    String content;      //영화 줄거리
    String poster;       //영화 포스터 (이미지)
    String logo;         //영화사 로고 (이미지)

    TextView txtTitle;
    TextView txDirector;
    TextView txGenre;
    TextView txGrade;
    TextView txPlaydate;
    TextView txContent;

```

```

NetworkImageView imPoster;

protected RequestQueue mQueue = null;    //인터넷에 뭔가 요청할때 사용하는것은 이 mQueue에 넣어줌.
protected ImageLoader mImageLoader = null; //네트워크 이미지를 쓸때 사용하는 변수

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_movie);

    //Volley를위한 설정
    Cache cache = new DiskBasedCache(getCacheDir(), 1024 * 1024);
    Network network = new BasicNetwork(new HurlStack());
    mQueue = new RequestQueue(cache, network);
    mQueue.start();

    mImageLoader = new ImageLoader(mQueue,
        new LruBitmapCache(LruBitmapCache.getCacheSize(this)));

    Intent intent = getIntent();    //MainActivity 데이터를 받음.
    data = intent.getStringArrayExtra("data");
    title = data[0];
    director = data[1];
    playdate = data[2];
    genre = data[3];
    grade = data[4];
    content = data[5];
    poster = data[6];
    logo = data[7];

    txtTitle = (TextView) findViewById(R.id.title);
    txtTitle.setText(title);

    txDirector = (TextView) findViewById(R.id.director);
    txDirector.setText(director);

    txGenre = (TextView) findViewById(R.id.genre);
    txGenre.setText(genre);

    txGrade = (TextView) findViewById(R.id.grade);
    txGrade.setText(grade);

    txPlaydate = (TextView) findViewById(R.id.playdate);
    txPlaydate.setText(playdate);

    txContent = (TextView) findViewById(R.id.content);
    txContent.setText(content);

```

```

        imPoster = (NetworkImageView) findViewById(R.id.poster);
        imPoster.setImageUrl("http://192.168.0.2/customlist/"
            + poster, mImageLoader);

    }
}

```

activity_movie.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:paddingBottom="5dp"
        android:paddingLeft="5dp"
        android:paddingRight="5dp"
        android:paddingTop="5dp">

        <TextView
            android:id="@+id/title"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="3dp"
            android:gravity="center"
            android:shadowColor="#FF464646"
            android:shadowDx="2.0"
            android:shadowDy="2.0"
            android:shadowRadius="1.0"
            android:textColor="#81F781"
            android:textSize="35dp" />

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="3dp"
            android:orientation="horizontal"
            android:padding="5dp">

            <com.android.volley.toolbox.NetworkImageView
                android:id="@+id/poster"
                android:layout_width="150dp"
                android:layout_height="200dp"

```

```
        android:layout_alignParentLeft="true"
        android:padding="5dp"
        android:scaleType="centerCrop" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="3dp"
    android:orientation="vertical"
    android:padding="5dp">
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="장르"
        android:textSize="16sp" />
```

```
    <TextView
        android:id="@+id/genre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#00f"
        android:textSize="16sp" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="감독"
        android:textSize="16sp" />
```

```
    <TextView
        android:id="@+id/director"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#00f"
        android:textSize="16sp" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="등급"
        android:textSize="16sp" />
```

```
    <TextView
        android:id="@+id/grade"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:textColor="#00f"
        android:textSize="16sp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="개봉일"
            android:textSize="16sp" />

        <TextView
            android:id="@+id/playdate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#00f"
            android:textSize="16sp" />

    </LinearLayout>
</LinearLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="3dp"
    android:gravity="left"
    android:shadowColor="#FF464646"
    android:shadowDx="2.0"
    android:shadowDy="2.0"
    android:shadowRadius="1.0"
    android:text="줄거리"
    android:textColor="#81F781"
    android:textSize="20dp" />

<TextView
    android:id="@+id/content"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#00f"
    android:textSize="15sp" />

</LinearLayout>
</ScrollView>

```

selectjson.php

```
<?php
$dbc = mysqli_connect('127.0.0.1', 'root', 'apmsetup', 'MovieDB')
    or die('Error Connecting to MySQL server.');
```

mysqli_query(\$dbc, "set names utf8;");

\$query = "select * from movie";

\$result = mysqli_query(\$dbc, \$query)
 or die('Error Querying database.');

\$json = array();
if(mysqli_num_rows(\$result)){
 while(\$row=mysqli_fetch_assoc(\$result)){
 \$json['list'][]= \$row;
 }
}

echo json_encode(\$json);

mysqli_free_result(\$result);

mysqli_close(\$dbc);
?>

MySQL DB Query

```
drop database if exists MovieDB;
create database MovieDB DEFAULT CHARACTER SET utf8 collate utf8_general_ci;
use MovieDB;
set names utf8;
```

create table movie(
 id int(11) not null auto_increment,
 title varchar(50),
 director varchar(30),
 playdate varchar(255),
 poster varchar(255),
 logo varchar(255),
 grade varchar(50),
 genre varchar(50),
 content varchar(1000),
 primary key (id),
 index(title)
) ENGINE=InnoDB default character set utf8 collate utf8_general_ci;

set names euckr;

3 실행 결과

겨울 왕국
감독: 크리스 벅
개봉일: 2014.01.16 개봉

범죄와의 전쟁
감독: 윤종빈
개봉일: 2012.02.02 개봉

어벤져스
감독: 조스 웨던
개봉일: 2012.04.26 개봉

수어사이드 스쿼드
감독: 데이비드 에이어
개봉일: 2016.08.03 개봉

인셉션
감독: 크리스토퍼 놀란
개봉일: 2010.07.21 개봉

해리포터 죽음의 성물
감독: 데이빗 예이츠
개봉일: 2011.07.13 개봉

인셉션
감독: 크리스토퍼 놀란
개봉일: 2010.07.21 개봉

해리포터 죽음의 성물
감독: 데이빗 예이츠
개봉일: 2011.07.13 개봉

용서는 없다
감독: 김형준
개봉일: 2010.01.07 개봉

데드풀
감독: 팀 밀러
개봉일: 2016.02.17 개봉

인터스텔라
감독: 크리스토퍼 놀란
개봉일: 2014.11.06 개봉

레드 라이트
감독: 로드리고 코르테스
개봉일: 2012.08.23 개봉

TestList

어벤져스

장르
액션/SF/모험
감독
조스 웨던
등급
12세 관람가
개봉일
2012.04.26 개봉

즐거리

ASSEMBLE! 최강의 슈퍼히어로들이 모였다!
지구의 운명을 건 거대한 전쟁이 시작된다!

지구의 안보가 위협당하는 위기의 상황에서
슈퍼히어로들을 불러모아 세상을 구하는, 일명 [어벤져스]
작전.
에너지원 '큐브'를 이용한 적의 등장으로 인류가 위험에
처하자 국제평화유지기구인 쉴드 (S.H.I.E.L.D)의 국장 닉
퓨리(사무엘 L. 잭슨)는 [어벤져스] 작전을 위해 전 세계에
흩어져 있던 슈퍼히어로들을 찾아나선다. 아이언맨(로버트
다우니 주니어)부터 토르(크리스 헬스워스), 헐크(마크
러팔로), 캡틴 아메리카(크리스 에반스)는 물론, 쉴드의
요원인 블랙 위도우(스칼렛 요한슨), 호크 아이(제레미
레너)까지, 최고의 슈퍼히어로들이 [어벤져스]의 멤버로
모이게 되지만 각기 개성이 강한 이들의 만남은 예상치

TestList

겨울 왕국

장르
애니메이션/모험/코미디/
가족/판타지/뮤지컬
감독
크리스 벅
등급
전체관람가
개봉일
2014.01.16 개봉

즐거리

얼어붙은 세상을 녹일 자매가 온다!

서로가 최고의 친구였던 자매 '엘사'와 '안나'. 하지만 언니
'엘사'에게는 하나뿐인 동생에게조차 말 못할 비밀이 있다.
모든 것을 얼려버리는 신비로운 힘이 바로 그것. '엘사'는
통제할 수 없는 자신의 힘이 두려워 왕국을 떠나고,
얼어버린 왕국의 저주를 풀기 위해 '안나'는 언니를 찾아
환상적인 여정을 떠나는데.....

TestList

인터스텔라

장르
SF
감독
크리스토퍼 놀란
등급
12세 관람가
개봉일
2014.11.06 개봉

즐거리

"우린 답을 찾을 거야, 늘 그랬듯이"

세계 각국의 정부와 경제가 완전히 붕괴된 미래가
다가온다.
지난 20세기에 범한 잘못이 전 세계적인 식량 부족을
불러왔고, NASA도 해체되었다.
이때 시공간에 불가사의한 틈이 열리고, 남은 자들에게는
이 곳을 탐험해 인류를 구해야 하는 임무가 지원된다.
사랑하는 가족들을 뒤로 한 채 인류라는 더 큰 가족을 위해,
그들은 이제 희망을 찾아 우주로 간다.
그리고 우린 답을 찾을 것이다. 늘 그랬듯이...

4 과제 수행시 어려웠던 점과 해결방법

커스텀 리스트가 이전에 해왔던 리스트와 크게 다른점은 없었기 때문에 큰 어려움은 없었던 것 같습니다. 다만, 처음에 아이템을 클릭한 뒤, 새로운 액티비티로 데이터를 넘겨주는 작업을 수행했을 때, 이미지의 경우엔 Volley를 사용해야 하기 때문에, 새로운 액티비티에서는 이를 다시 설정해주는 작업이 필요했는데 이를 설정하지 않아서, 처음에 새로운 액티비티로 이미지가 넘어오지 않아 많은 시간을 소요했던 것 같습니다.