

## 모바일 프로그래밍 실습 보고서

학번	201411203	이름	박소영
순서	12	날짜	2018.04.19 목요일
강의자료	06 - 화면전환		
주제	인텐트 필터, 알람, 타이머		

- **인텐트 필터 : 내 앱을 묵시적인 방법으로 찾아내는 수단!**

- 액티비티에서 인텐트는 많이 발생한다.  
그 중에서 내 조건을 만족하는, 필터를 만족하는 것만 내 앱을 실행시킬 수 있다!
- 인텐트 중에서 나와 적합한 것을 가려내는 것.
- 프로그램의 Manifest 파일에 등록된다!
- 인텐트 필터는 한가지만 있는 게 아니라, 여러 개 만들어질 수도 있다.
  - ➔ 여기 필터 중에 **하나만 걸리면** 이 앱이 호출되게 된다.
- **Action** : 액션을 검사한다.
- **Category** : 카테고리를 꼭 지정해줘야 한다! 없으면 디폴트라도 설정해야 한다.
- **Data** : 데이터도 필터링 하고 싶으면 데이터에 대한 URI 정보와 타입 정보를 넣어준다.

- **컨텐츠 프로바이더** : 안드로이드 내의 데이터를 제공하고 있어서 이걸 갖다가 쓰면 된다.

- 인텐트를 실행시키고 화면을 전환한 다음에 결과를 받고 싶으므로

- 인텐트를 생성할 때 클래스명을 쓰지 않고 타입 이름을 준다.
- startActivityForResult 함수를 쓴다.

- **PendingIntent** : 인텐트 정보를 가지고 있다. 바로 화면 전환을 하지 않고 대기.

- 특정 상태가 되면 특정 액티비티로 연결한다.
- 데이터나 컴포넌트에 관련된 정보를 저장해놓을 수 있다.
- param : 코드 정보, 사용할 인텐트 정보(어디서 어디로 이동할 건지), 플래그 값 (Lec06\_25p)
- 언제 쓸까?
  1. 사용자가 Notification을 통해 특정한 동작을 할 때, 실행되는 인텐트를 생성함 (NotificationManager가 인텐트를 실행)
  2. 사용자가 AppWidget을 통해 특정한 동작을 할 때, 실행되는 인텐트를 생성함 (홈 스크린이 인텐트를 실행)
  3. 미래 특정 시점에 실행되는 인텐트 선언 (안드로이드의 AlarmManager가 인텐트를 실행)

```
getActivitiy (Context context, int requestCode, Intent intent, int flags);
```

+ flags : FLAG\_ONE\_SHOT, FLAG\_NO\_CREATE, FLAG\_CANCEL\_CURRENT, FLAG\_UPDATE\_CURRENT

## AndroidManifest.xml <프로젝트 1>

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="moblieprogramming.a20180419">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.PICK" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
            <intent-filter>
                <action android:name="mobileprogramming.a20180419.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

## java 소스코드 : MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    final int SELECT_IMAGE = 100;

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

```

        imageView = (ImageView) findViewById(R.id.imageView);
    }
    boolean checkAppPermission(String[] requestPermission){
        boolean[] requestResult = new boolean[requestPermission.length];
        for(int i=0; i< requestResult.length; i++){
            requestResult[i] = (ContextCompat.checkSelfPermission (this,
                requestPermission[i]) == PackageManager.PERMISSION_GRANTED);
            if(!requestResult[i]){
                return false;
            }
        }
        return true;
    }

    void askPermission(String[] requestPermission, int REQ_PERMISSION) {
        ActivityCompat.requestPermissions(this, requestPermission,
            REQ_PERMISSION);
    }

    @Override
    public void onRequestPermissionsResult (int requestCode,
        @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
            grantResults);
        switch (requestCode) {
            case SELECT_IMAGE :
                if (checkAppPermission(permissions)) {
                    Intent intent = new Intent(Intent.ACTION_PICK);
                    intent.setType(MediaStore.Images.Media.CONTENT_TYPE);
                    intent.setData(MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                    startActivityForResult(intent, SELECT_IMAGE);
                } else {
                    Toast.makeText(getApplicationContext(),
                        "권한이 거절되었습니다.", Toast.LENGTH_SHORT).show();
                    finish(); // 앱을 중지시키는 함수
                }
                break;
        }
    }

    public void btnClick(View view) {
        if (checkAppPermission(new String[]
            {Manifest.permission.READ_EXTERNAL_STORAGE})) {

```

```

        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType(MediaStore.Images.Media.CONTENT_TYPE);
        intent.setData(MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(intent, SELECT_IMAGE);
    } else {
        askPermission(new String[]
            {Manifest.permission.READ_EXTERNAL_STORAGE}, SELECT_IMAGE);
    }
}

// 데이터를 받으면 자동으로 이 함수가 호출된다.
@Override
protected void onActivityResult(int requestCode,
    int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == SELECT_IMAGE) {
        if (resultCode == Activity.RESULT_OK) {
            /* Bitmap bmp = null;
            try {
                // 실제 데이터의 위치값에 해당하는 Path 정보를 받아온다.
                bmp =
MediaStore.Images.Media.getBitmap(getContentResolver(), data.getData());
                imageView.setImageBitmap(bmp);
            } catch (IOException e) {
                e.printStackTrace();
            } */

            imageView.setImageURI(data.getData());
            // 인텐트로부터 들어온 정보를 바로 줄 수도 있다.

        }
    }
}
}

```

## java 소스코드 : MainActivity.java (프로젝트 2)

```

public class MainActivity extends AppCompatActivity {
    TimerTask timerTask;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

TimerTask timerTask = new TimerTask() {
    @Override
    public void run() {
        makeNotification();
    }
};
Timer timer = new Timer();
timer.schedule(timerTask, 1000, 1000);
}

public void makeNotification() {
    NotificationManager manager = (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);
    Notification.Builder builder = new Notification.Builder(this);
    builder.setSmallIcon(R.mipmap.ic_launcher);
    builder.setLargeIcon(BitmapFactory.decodeResource(getResources(),
        R.mipmap.ic_launcher));
    builder.setTicker("알람");
    builder.setContentTitle("알람");
    builder.setContentText("일어나세요!!!");
    builder.setVibrate(new long[]{0,3000});
    Uri soundUri= RingtoneManager.getActualDefaultRingtoneUri(this,
        RingtoneManager.TYPE_NOTIFICATION);
    builder.setSound(soundUri);
    builder.setAutoCancel(true);

    Intent intent = new Intent(this, MainActivity.class);
    PendingIntent pIntent = PendingIntent.getActivity(this, 1, intent,
        PendingIntent.FLAG_UPDATE_CURRENT);
    builder.setContentIntent(pIntent);

    Notification notification= builder.build();
    manager.notify(1, notification);
}

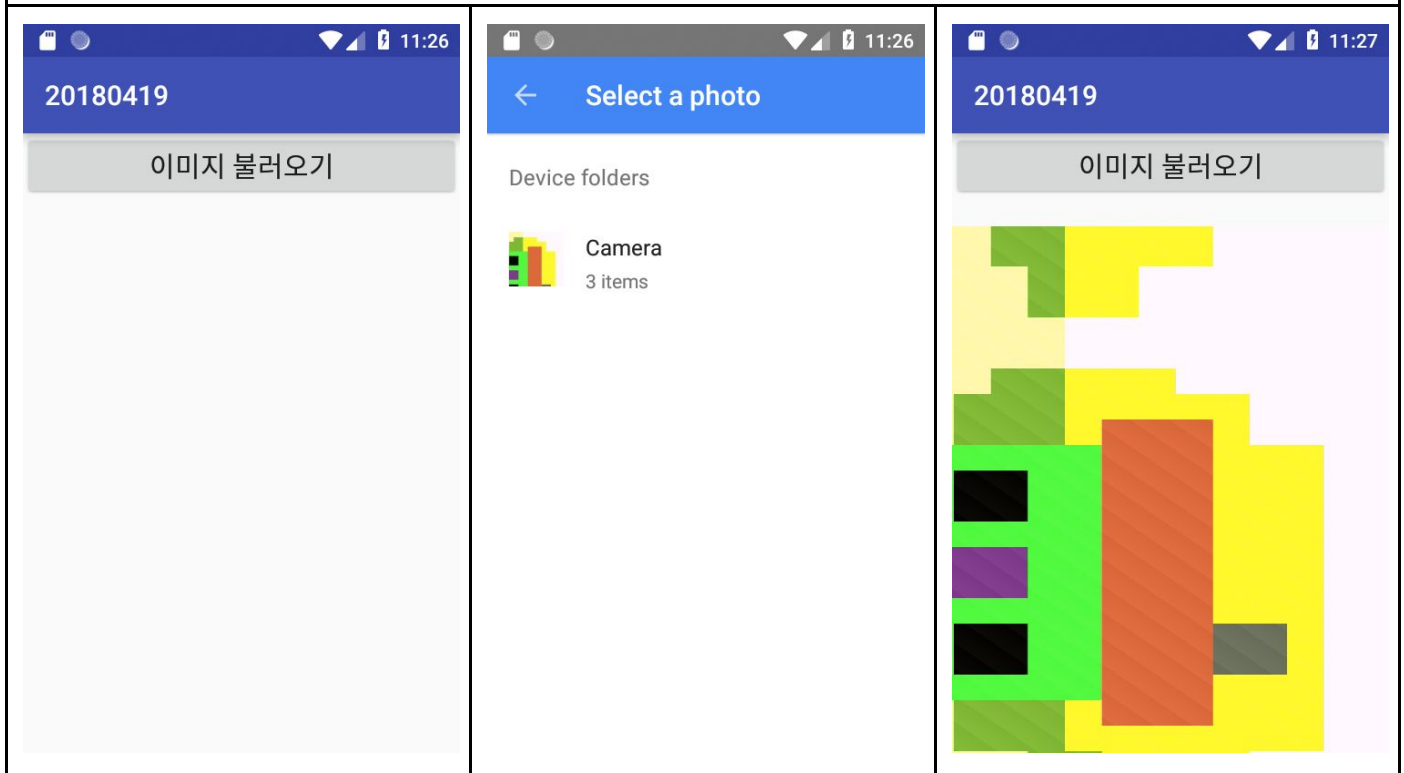
public void btnClick(View view) {
    // 버튼을 누르면 ACTION_PICK에 걸려있는 앱들 중에서 선택이 가능하게 된다.
    Intent intent = new Intent(Intent.ACTION_PICK);
    startActivity(intent);
}

```

}

## 실행 결과

### <프로젝트 1>



### <프로젝트 2>

