

208

액티비티 간 데이터 주고받기

실행되는 액티비티에 데이터 전달하기

209

- Application name: ActivityDataCom
 - ▣ Company Domain: company.co.kr
 - ▣ Package name: kr.co.company.activitydatacom

- ▣ ActivityA.java
- ▣ activity_a.xml

- ▣ ActivityB.java
- ▣ activity_b.xml

실행되는 액티비티에 데이터 전달하기

210

□ activity_a.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding = "20dp"
    android:orientation="vertical">

    <Button
        android:id="@+id/load_pic_url_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"/>

</LinearLayout>
```

실행되는 액티비티에 데이터 전달하기

211

□ activity_b.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding = "20dp"
    android:orientation="vertical">

    <TextView
        android:id="@+id/received_pic_folder_url"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

실행되는 액티비티에 데이터 전달하기

212

□ ActivityA.java

```
public class ActivityA extends AppCompatActivity {
```

```
    private String mSendPicUrl = "/data/pics";
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_a);
```

```
        // 버튼에 "사진 불러오기 : " 와 불러올 폴더 명을 출력한다.
```

```
        // =====
```

```
        Button loadPicUrlBtn = (Button) findViewById(R.id.load_pic_url_btn);
```

```
        String loadUrlStr = "사진 불러오기 : " + mSendPicUrl;
```

```
        loadPicUrlBtn.setText(loadUrlStr);
```

```
        // =====
```

```
    }
```

실행되는 액티비티에 데이터 전달하기

213

□ ActivityA.java

A → B

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.load_pic_url_btn:  
            // 1. 사진 불러오기 액티비티를 실행하기 위한 인텐트를 생성한다.  
            // =====  
            Intent intent = new Intent();  
            // =====  
  
            // 2. 명시적 인텐트 방식으로 실행될 B 액티비티를 설정한다.  
            // =====  
            intent.setClass(this, ActivityB.class);  
            // =====  
  
            // 3. 불러올 사진 폴더 이름을 인텐트 엑스트에 추가한다.  
            // =====  
            intent.putExtra("PIC_URL", mSendPicUrl);  
            // =====  
  
            // 4. 사진 불러오기 B 액티비티를 실행한다.  
            // =====  
            startActivityForResult(intent, 0);  
            // =====  
        }  
    }  
}
```

request code
↳ 다른 Activity 구분

실행되는 액티비티에 데이터 전달하기

214

□ ActivityB.java

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_b);
```

```
    // 1. A 액티비티로부터 전달 받은 인텐트를 참조한다.
```

```
    // =====  
    Intent intent = getIntent();  
    // =====
```

```
    // 2. 인텐트에 포함된 엑스트라에서 사진 폴더명을 추출한다.
```

```
    // =====  
    Bundle bundle = intent.getExtras();  
    String picUrl = bundle.getString("PIC_URL");  
    // =====
```

```
    // 3. 텍스트뷰에 "전달 받은 사진 폴더 : " 와 불러올 폴더 명을 출력한다.
```

```
    // =====  
    TextView loadPicUrlTextView =  
        (TextView) findViewById(R.id.received_pic_folder_url);  
    String picUrlStr = "전달 받은 사진 폴더 : " + picUrl;  
    loadPicUrlTextView.setText(picUrlStr);  
    // =====
```

```
}
```

실행되는 액티비티에 데이터 전달하기

215

□ AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.activitydatacom">

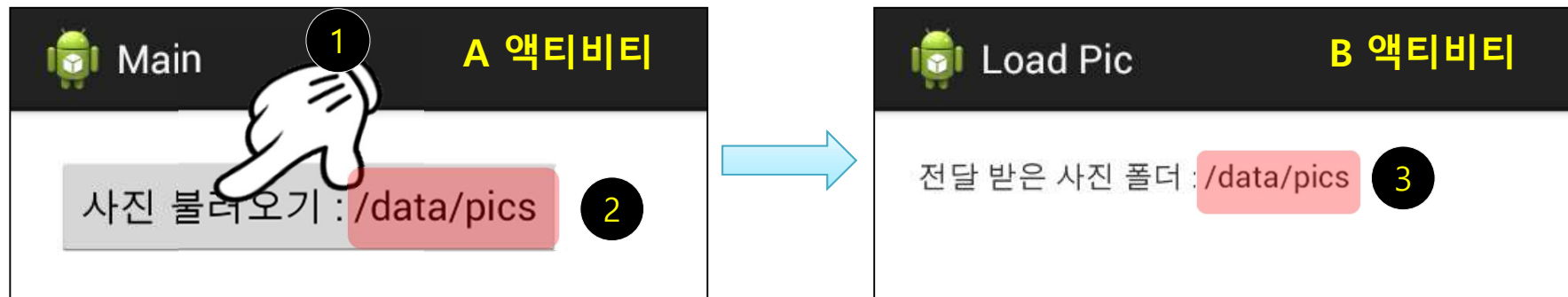
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ActivityA" android:label="Main">
            <intent-filter>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

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ActivityB" android:label="Load Pic"/>
    </application>

</manifest>
```

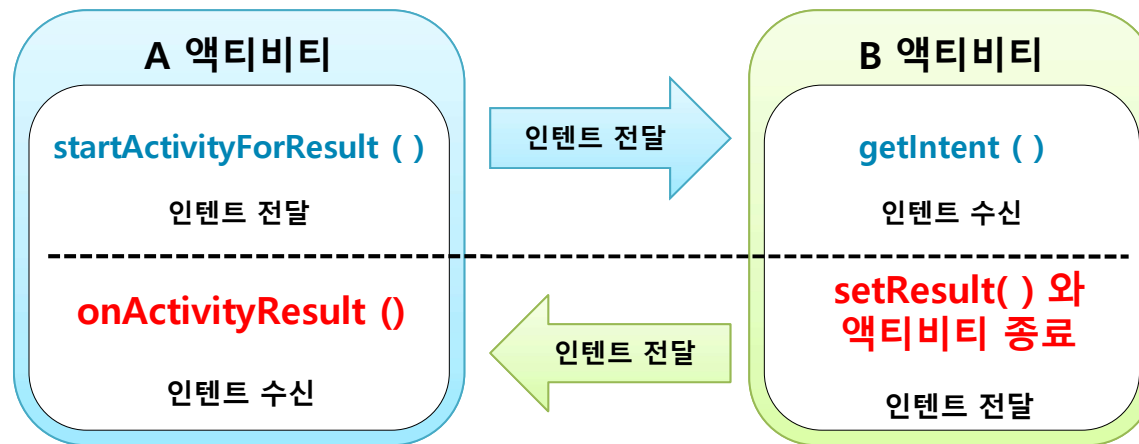

실행되는 액티비티에 데이터 전달하기

216



실행된 액티비티가 실행한 액티비티에 인텐트 전달

217



실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비의 결과받기 과정

218

- activity_b.xml
 - ▣ 버튼 추가

<Button

```
android:id="@+id/send_pic_file_url_btn"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="onClick"/>
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비의 결과받기 과정

219

□ ActivityB.java

```
public class ActivityB extends AppCompatActivity {
```

```
    private String mResultPicFileUrl = "/data/pics/a.png";
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_b);
```

```
        // 1. A 액티비티로부터 전달 받은 인텐트를 참조한다.
```

```
        // =====  
        Intent intent = getIntent();  
        // =====
```

```
        // 2. 인텐트에 포함된 엑스트라에서 사진 폴더명을 추출한다.
```

```
        // =====  
        Bundle bundle = intent.getExtras();  
        String picUrl = bundle.getString("PIC_URL");  
        // =====
```

```
        // 3. 텍스트뷰에 "전달 받은 사진 폴더 : " 와 불러올 폴더 명을 출력한다.
```

```
        // =====  
        TextView loadPicUrlTextView =  
            (TextView) findViewById(R.id.received_pic_folder_url);  
        String picUrlStr = "전달 받은 사진 폴더 : " + picUrl;  
        loadPicUrlTextView.setText(picUrlStr);  
        // =====
```

```
        // 4. 버튼에 "선택된 사진 경로 전달 : " 과 선택된 사진 경로를 출력한다.
```

```
        // =====  
        Button resultPicFileUrlBtn =  
            (Button) findViewById(R.id.send_pic_file_url_btn);  
        String picFileUrlStr = "선택된 사진 경로 전달 \n" + mResultPicFileUrl;  
        resultPicFileUrlBtn.setText(picFileUrlStr);  
        // =====  
    }
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비의 결과받기 과정

220

□ ActivityB.java

```
public void onClick( View v ) {  
    // 선택된 사진 파일 경로를 전달할 인텐트 생성 및 엑스트라를 설정한다.  
    // =====  
    Intent intent = new Intent();  
    intent.putExtra("RESULT_PIC_FILE_URL", mResultPicFileUrl);  
    // =====  
  
    // 전달할 인텐트를 설정하고 액티비티를 종료한다.  
    // =====  
    setResult(RESULT_OK, intent);  
    finish();  
    // =====  
}  
}
```

result code

종료해달라

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비티의 결과받기 과정

221

프레임워크 소스 : Activity.java

```
public class Activity extends ContextThemeWrapper
    implements LayoutInflater.Factory2,
        Window.Callback, KeyEvent.Callback,
        OnCreateContextMenuListener, ComponentCallbacks2 {

    /** Standard activity result: operation canceled. */
    public static final int RESULT_CANCELED = 0;
    /** Standard activity result: operation succeeded. */
    public static final int RESULT_OK = -1;
    /** Start of user-defined activity results. */
    public static final int RESULT_FIRST_USER = 1;
```

커스텀 아이징
가능

2
3
4
5

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비티의 결과받기 과정

222

- activity_a.xml
 - ▣ 텍스트뷰 추가

<TextView

```
android:id="@+id/result_pic_file_url"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"/>
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비의 결과받기 과정

223

□ ActivityA.java

예약됨 (-1)

어떤 Activity의 결과인지 구분

예약 or 취소버튼이짐

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

if(resultCode == RESULT_OK) {

// 1. B 액티비티로부터 전달 받은 사진 파일 경로를 추출한다.

// =====

String picFileUrlStr =

data.getStringExtra("RESULT_PIC_FILE_URL");

// =====

// 2. 텍스트뷰에 사진 파일 경로를 출력한다.

// =====

TextView picFileUrlTv =

(TextView) findViewById(R.id.result_pic_file_url);

picFileUrlTv.setText("선택된 사진 파일 경로 : " + picFileUrlStr);

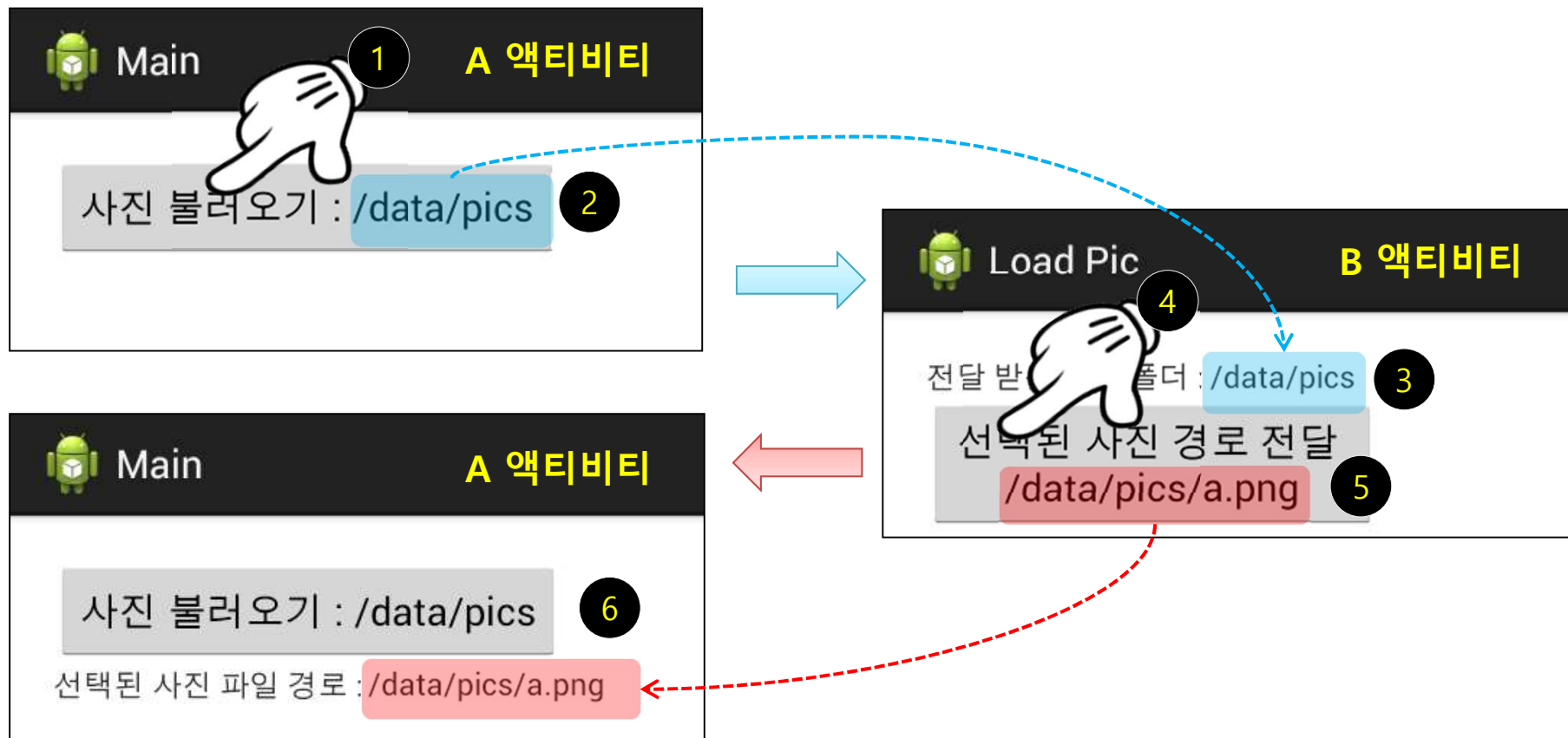
// =====

}

}

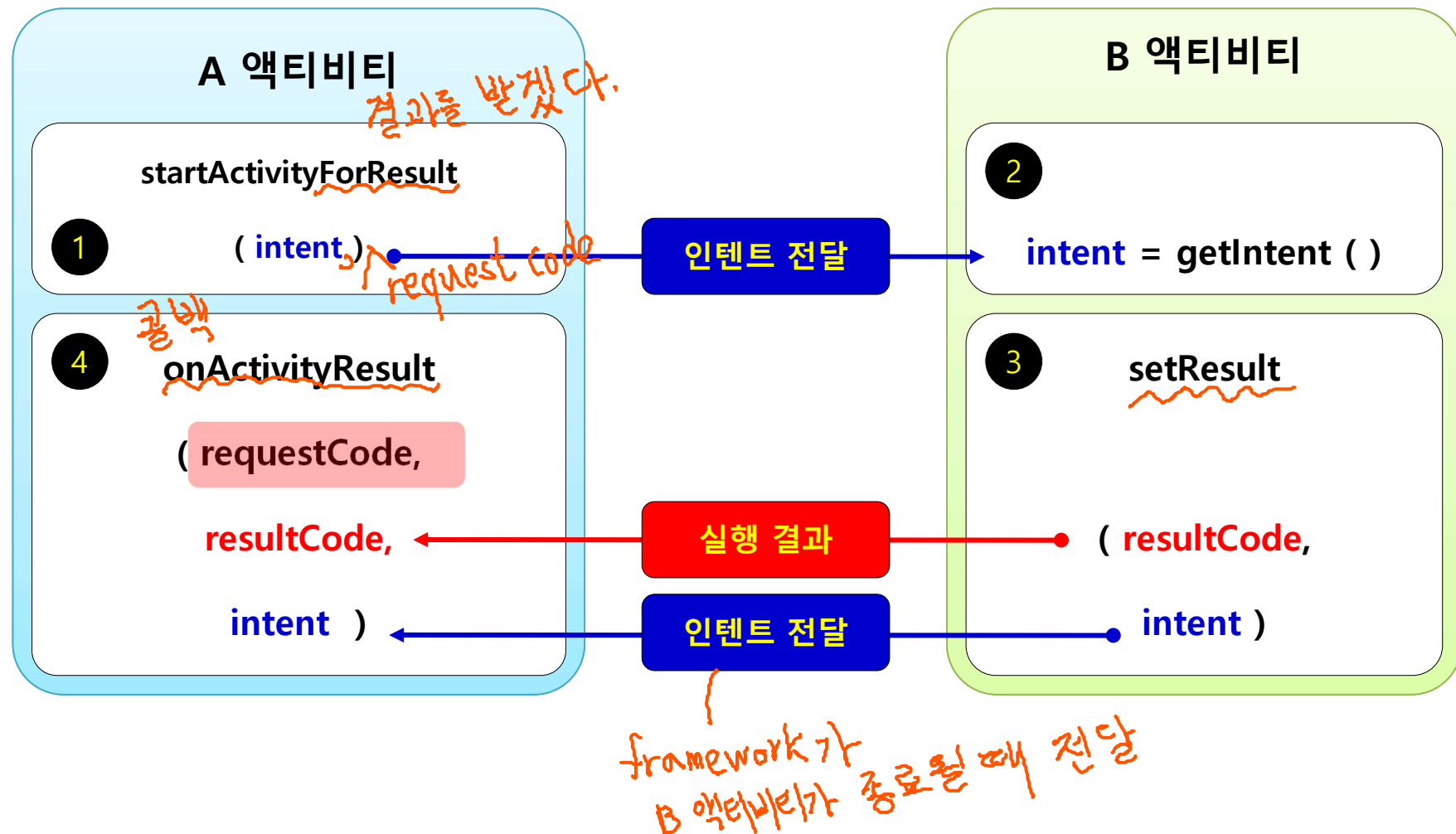
실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비의 결과받기 과정

224



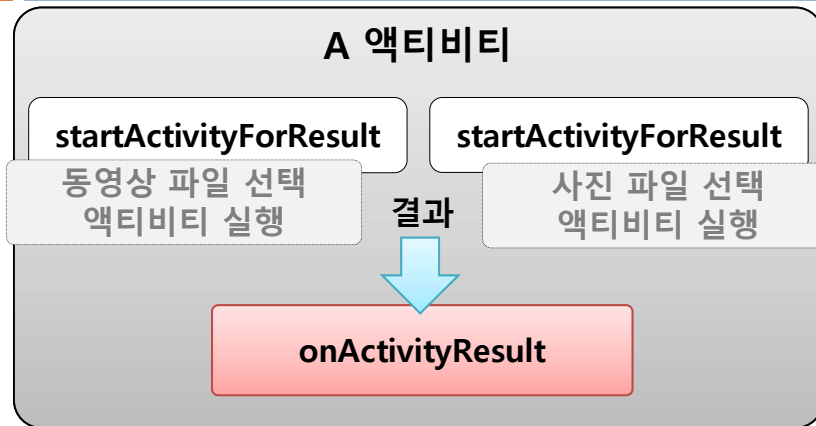
실행된 액티비티가 실행한 액티비티에 인텐트 전달 – 실행된 액티비의 결과받기 과정

225



실행된 액티비티가 실행한 액티비티에 인텐트 전달 – Request Code

226



앗 !!! onActivityResult
에서 어떤 액티비티의 결과인지
구별할 방법이 없네



아 Request Code로
구분하면 되겠구나!



사진 파일 선택
액티비티 결과 코드

동영상 파일 선택
액티비티 결과 코드



실행된 액티비티가 실행한 액티비티에 인텐트 전달 – Request Code

227

□ ActivityA.java

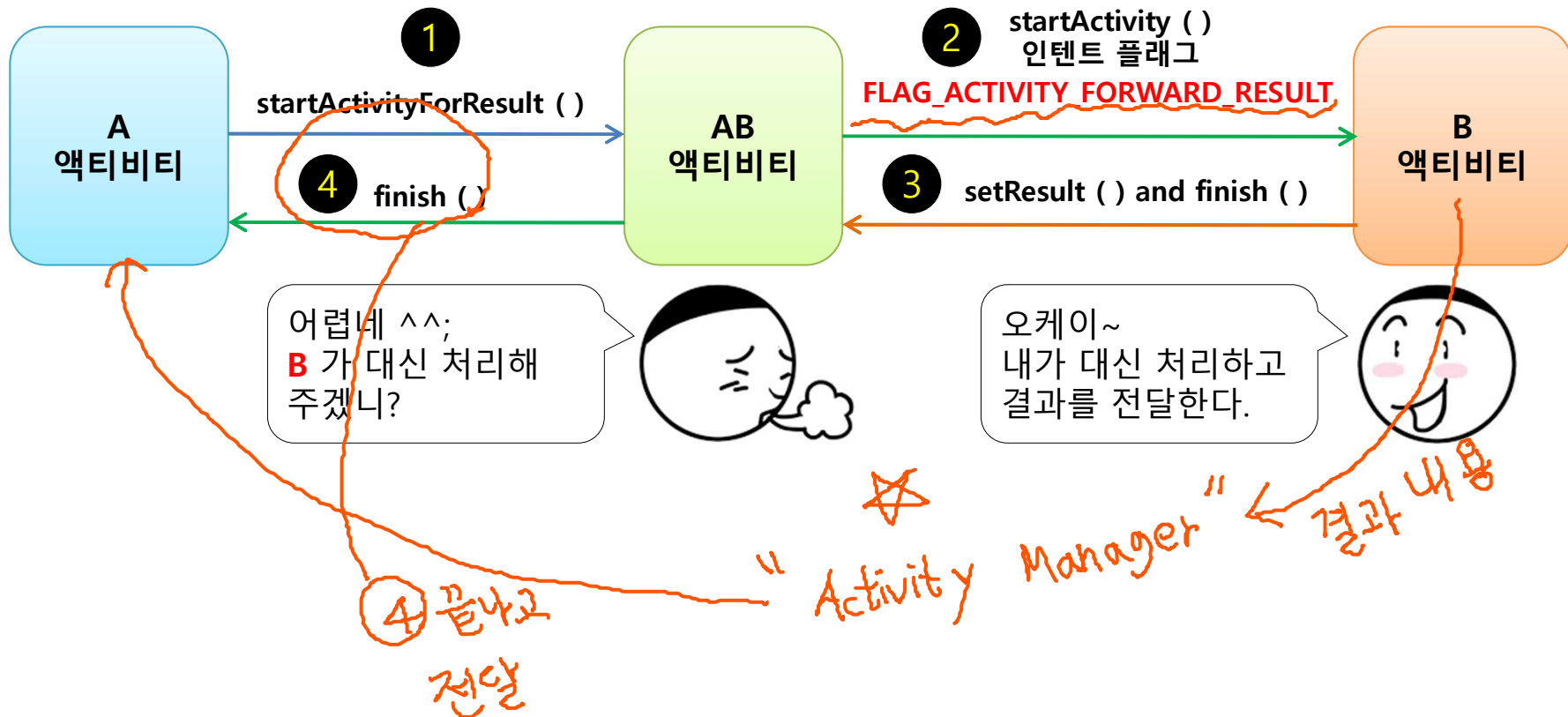
```
public class ActivityA extends AppCompatActivity {  
  
    ...  
  
    public static final int REQUEST_CODE_PIC_FILE_URL = 1;  
  
    ...  
  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.load_pic_url_btn:  
                ...  
                startActivityForResult(intent, REQUEST_CODE_PIC_FILE_URL);  
            }  
        }  
    }  
  
    @Override  
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
        if (requestCode == REQUEST_CODE_PIC_FILE_URL) {  
            if (resultCode == RESULT_OK) {  
                ...  
            }  
        }  
    }  
}
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 - FLAG_ACTIVITY_FORWARD_RESULT 인텐트 속성

228

flag

■ FLAG_ACTIVITY_FORWARD_RESULT 인텐트 플래그는 액티비티의 실행 결과를 다른 액티비티에 위임 할 수 있는 속성이다.



실행된 액티비티가 실행한 액티비티에 인텐트 전달 – FLAG_ACTIVITY_FORWARD_RESULT 인텐트 속성

229

□ activity_ab.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    android:orientation="vertical">

    <Button
        android:id="@+id/send_pic_file_url_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Forward Result > Activity B"
        android:onClick="onClick"/>

</LinearLayout>
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – FLAG_ACTIVITY_FORWARD_RESULT 인텐트 속성

230

□ ActivityAB.java

```
public class ActivityAB extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_ab);
```

```
    }
```

```
    public void onClick(View v) {
```

```
        // 1. B 액티비티가 실행되도록 전달받은 인텐트를 수정한다.
```

```
        // =====
```

```
        Intent intent = getIntent();
```

```
        intent.setClass(this, ActivityB.class);
```

```
        // =====
```

```
        // 2. B 액티비티가 대신 처리하도록 인텐트 플래그를 추가한다.
```

```
        // =====
```

```
        intent.addFlags(Intent.FLAG_ACTIVITY_FORWARD_RESULT);
```

```
        // =====
```

```
        startActivity(intent);
```

```
        // finish();
```

```
    }
```

```
}
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – FLAG_ACTIVITY_FORWARD_RESULT 인텐트 속성

231

- ActivityA.java
 - ▣ onClick 메소드 수정

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.load_pic_url_btn:  
            ...  
            intent.setClass(this, ActivityAB.class);  
            ...  
        }  
    }  
}
```


실행된 액티비티가 실행한 액티비티에 인텐트 전달 – FLAG_ACTIVITY_FORWARD_RESULT 인텐트 속성

232

□ AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.activitydatacom">

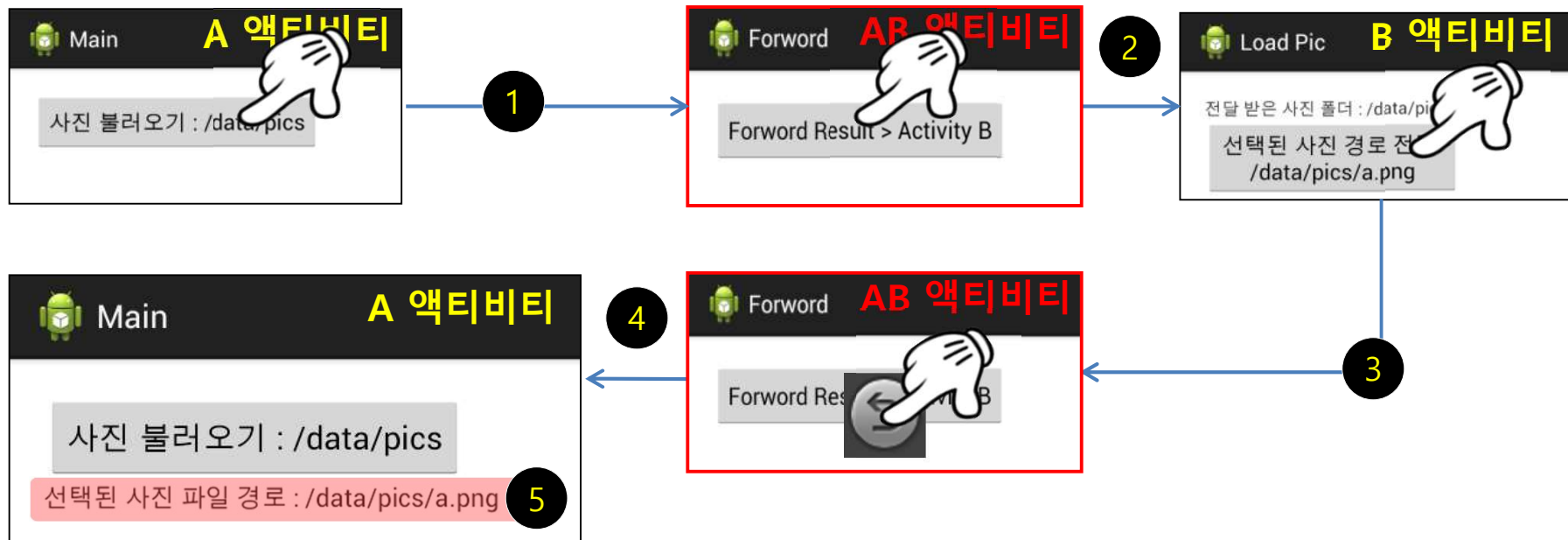
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ActivityA" android:label="Main">
            <intent-filter>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

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ActivityAB" android:label="Forword"/>
        <activity android:name=".ActivityB" android:label="Load Pic"/>
    </application>

</manifest>
```

실행된 액티비티가 실행한 액티비티에 인텐트 전달 – FLAG_ACTIVITY_FORWARD_RESULT 인텐트 속성

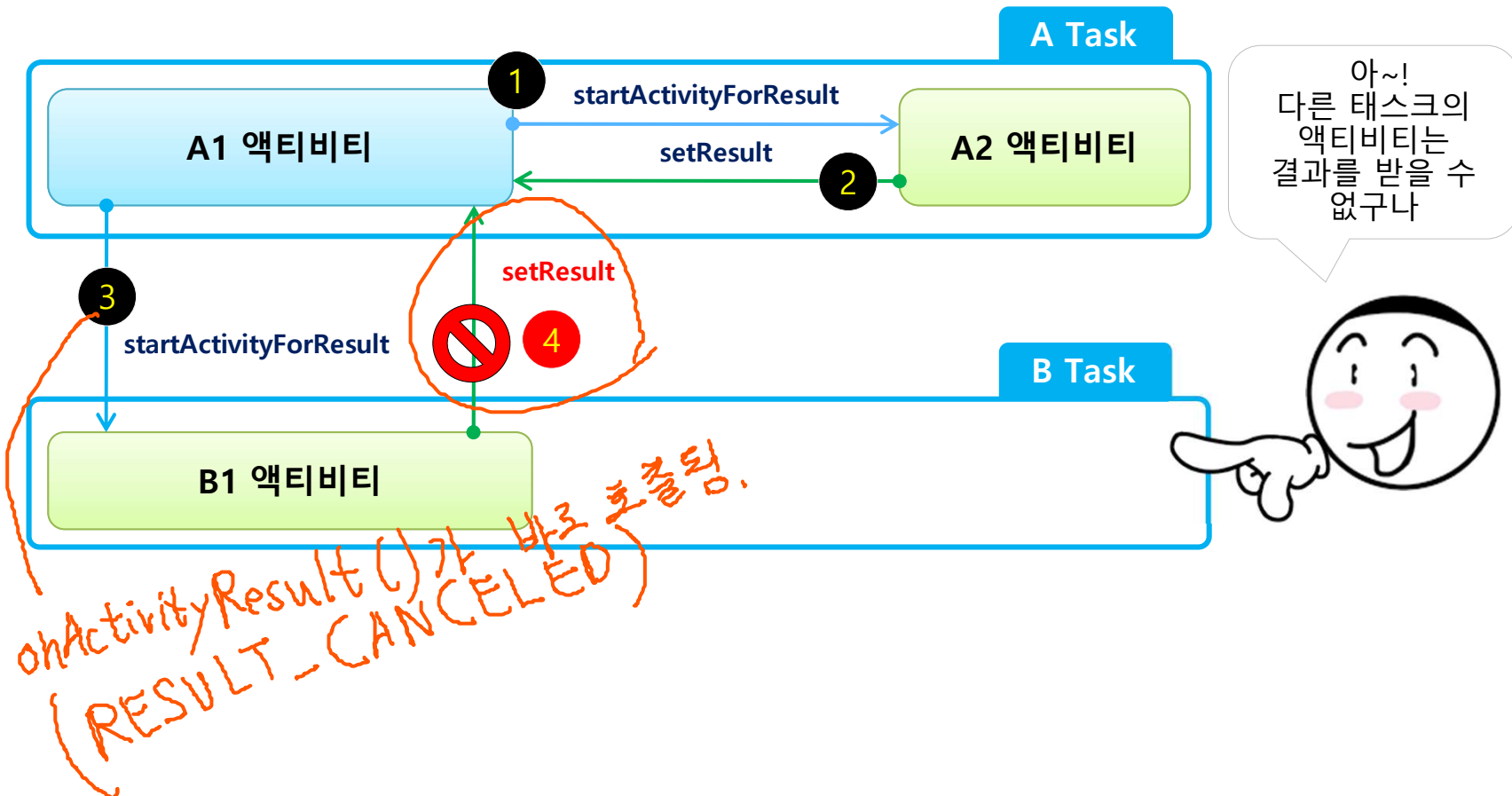
233



액티비티 간 데이터를 주고받을 때의 제약사항 -

서로 다른 태스크에서 실행된 액티비티 결과는 받을 수 없다

234



액티비티 간 데이터를 주고받을 때의 제약사항 – 서로 다른 태스크에서 실행된 액티비티 결과는 받을 수 없다

235

- ActivityA.java
 - ▣ onClick 메소드 수정

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.load_pic_url_btn:  
            ...  
            intent.setClass(this, ActivityB.class);  
            ...  
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |  
                             Intent.FLAG_ACTIVITY_MULTIPLE_TASK);  
            ...  
            startActivityForResult(intent, REQUEST_CODE_PIC_FILE_URL);  
        }  
    }  
}
```

액티비티 간 데이터를 주고받을 때의 제약사항 - 서로 다른 태스크에서 실행된 액티비티 결과는 받을 수 없다

236

- ActivityA.java
 - ▣ onActivityResult 메소드 수정

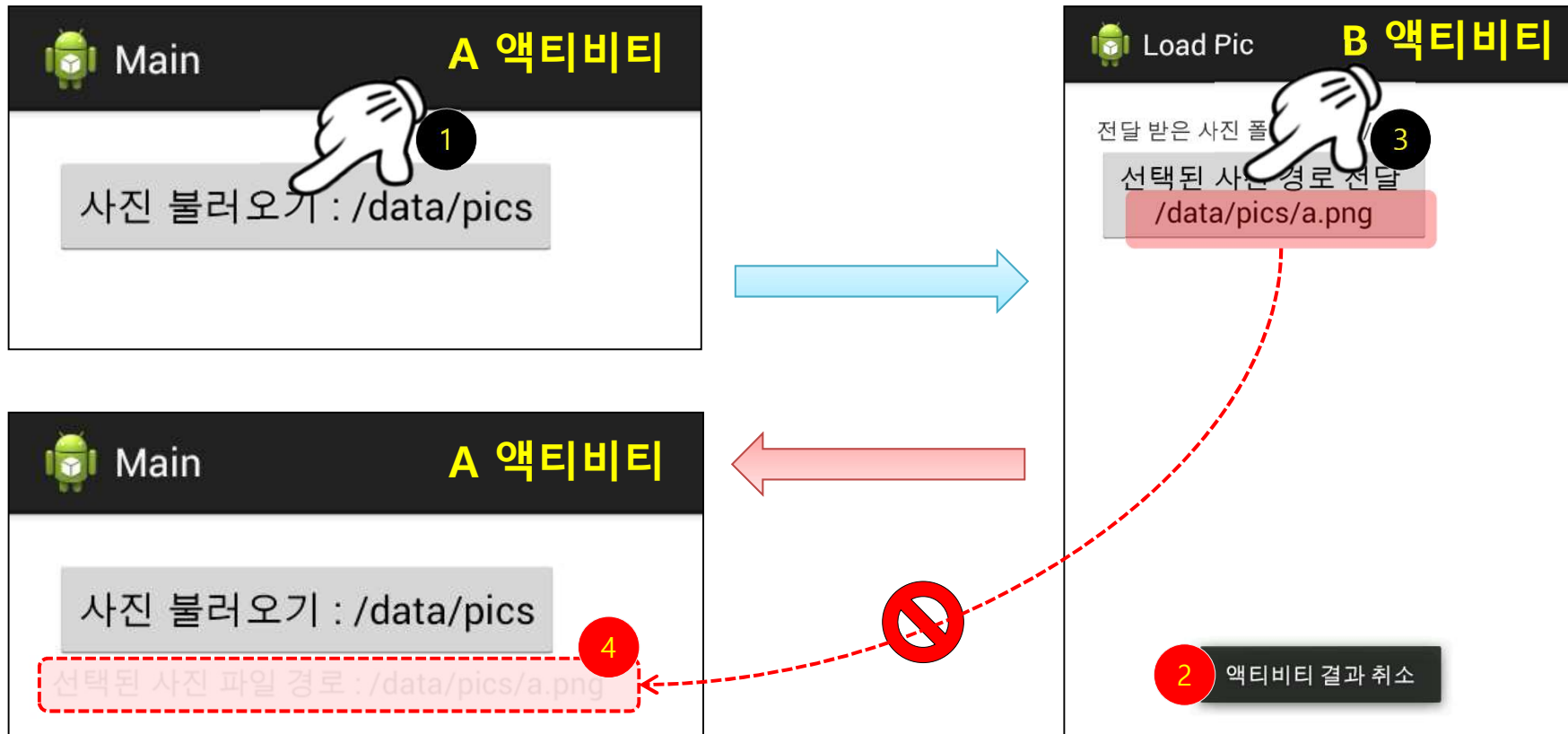
@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == REQUEST_CODE_PIC_FILE_URL) {  
        if (resultCode == RESULT_OK) {  
            ...  
        } else if (resultCode == RESULT_CANCELED) {  
            Toast.makeText(this, "액티비티 결과 취소",  
                Toast.LENGTH_LONG).show();  
        }  
    }  
}
```

액티비티 간 데이터를 주고받을 때의 제약사항 -

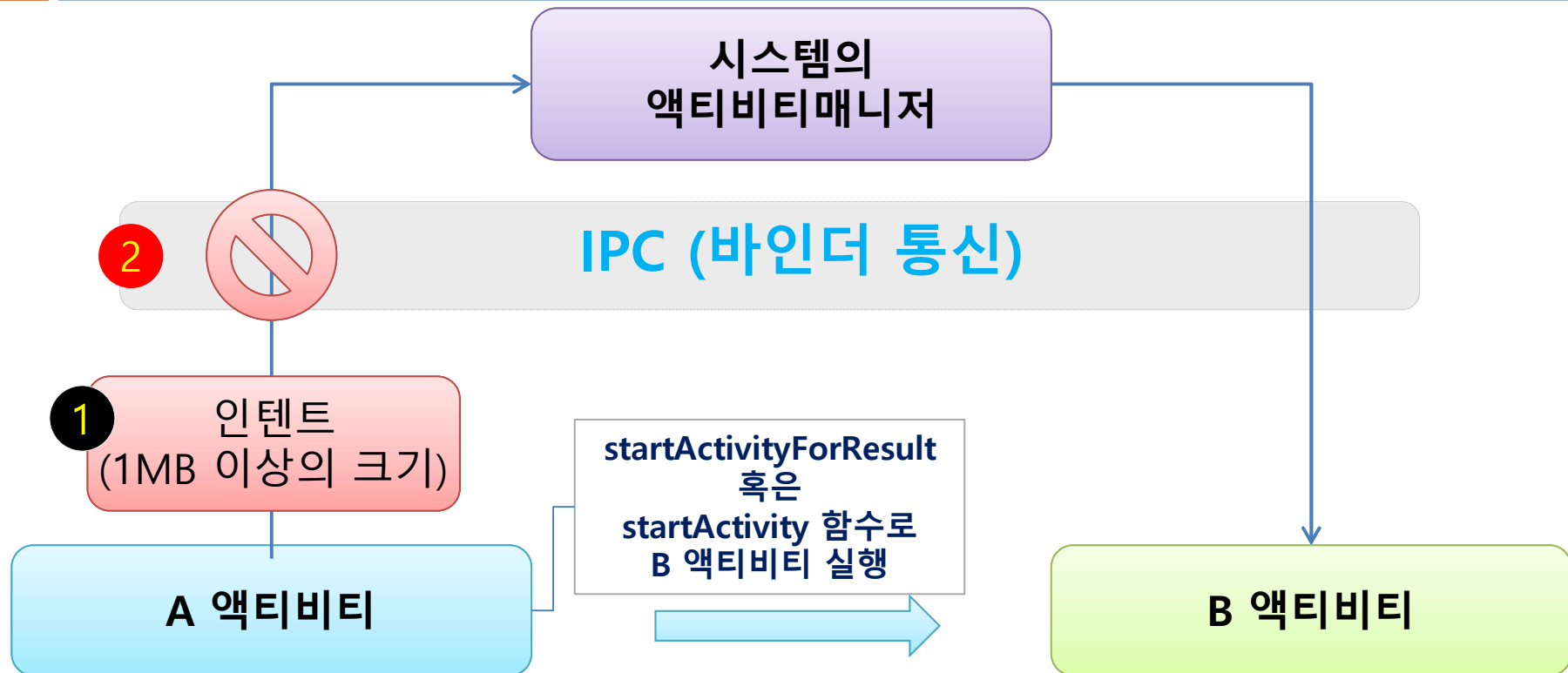
서로 다른 태스크에서 실행된 액티비티 결과는 받을 수 없다

237



액티비티 간 데이터를 주고받을 때의 제약사항 - 액티비티 간 주고받는 데이터는 100KB를 넘지 말자

238



프레임워크 소스 경로 : frameworks\base\libs\binder\ProcessState.cpp

`#define BINDER_VM_SIZE ((1*1024*1024) - (4096*2))`

액티비티 간 데이터를 주고받을 때의 제약사항 – 액티비티 간 주고받는 데이터는 100KB를 넘지 말자

239

■ 그렇다면 1MB 이내 인텐트 객체 크기라면 무조건 전송이 될까? 아쉽게도 그렇지 않다. 해당 버퍼는 여러 프로세스들이 공유해서 사용하기 때문에 1MB를 늘 보장받을 수 없다. 그러므로 안드로이드에서는 **100KB** 이내로 사용하기를 권장하고 있다.

액티비티 간 데이터를 주고받을 때의 제약사항 – 액티비티 간 주고받는 데이터는 100KB를 넘지 말자

240

□ ActivityA.java

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.load_pic_url_btn:  
            // 1. 사진 불러오기 액티비티를 실행하기 위한 인텐트를 생성한다.  
            // =====  
            Intent intent = new Intent();  
            // =====  
  
            // 2. 명시적 인텐트 방식으로 실행될 B 액티비티를 설정한다.  
            // =====  
            intent.setClass(this, ActivityB.class);  
            // =====  
  
            // 3. 불러올 사진 폴더 이름을 인텐트 엑스트에 추가한다.  
            // =====  
            intent.putExtra("PIC_URL", mSendPicUrl);  
            // =====  
  
            // 4. 엑스트라에 1 MB 이상의 데이터 전송  
            // =====  
            byte[] byte1mb = new byte[ 1024 * 1024];  
            intent.putExtra( "DATA_1MB", byte1mb );  
            // =====  
  
            // 5. 사진 불러오기 B 액티비티를 실행한다.  
            // =====  
            startActivityForResult(intent, REQUEST_CODE_PIC_FILE_URL);  
            // =====  
        }  
    }  
}
```

액티비티 간 데이터를 주고받을 때의 제약사항 – 액티비티 간 주고받는 데이터는 100KB를 넘지 말자

241



액티비티 간 데이터를 주고받을 때의 제약사항 - 액티비티 간 주고받는 데이터는 100KB를 넘지 말자

242

```
Run: AVD: Nexus_S_API_23 app
E/EGL_emulation: tid 4087: eglSurfaceAttrib(1165): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xa3c34d40, error=EGL_BAD_MATCH
E/JavaBinder: !!! FAILED BINDER TRANSACTION !!! (parcel size = 1049048)
D/AndroidRuntime: Shutting down VM
E/AndroidRuntime: FATAL EXCEPTION: main
at android.view.View$PerformClick.run(View.java:21147)
at android.os.Handler.handleCallback(Handler.java:739)
at android.os.Handler.dispatchMessage(Handler.java:95)
at android.os.Looper.loop(Looper.java:148)
at android.app.ActivityThread.main(ActivityThread.java:5417)
at java.lang.reflect.Method.invoke(Native Method) <2 more...>
Caused by: android.os.TransactionTooLargeException: data parcel size 1049048 bytes
at android.os.BinderProxy.transactNative(Native Method)
at android.os.BinderProxy.transact(Binder.java:503)
at android.app.ActivityManagerProxy.startActivity(ActivityManagerNative.java:2657)
at android.app.Instrumentation.execStartActivity(Instrumentation.java:1507)
at android.app.Activity.startActivityForResult(Activity.java:3917)
at android.support.v4.app.BaseFragmentActivityJB.startActivityForResult(BaseFragmentActivityJB.java:48)
```

클릭 통신은 parcel 객체

액티비티 간 데이터를 주고받을 때의 제약사항 -

생명주기 onPause, onStop, onDestroy에서는 setResult 함수를 사용할 수 없다

243

"Resumed"
상태에서 호출해야 함.

□ ActivityA.java

```
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.load_pic_url_btn:
            // 1. 사진 불러오기 액티비티를 실행하기 위한 인텐트를 생성한다.
            // =====
            Intent intent = new Intent();
            // =====

            // 2. 명시적 인텐트 방식으로 실행될 B 액티비티를 설정한다.
            // =====
            intent.setClass(this, ActivityB.class);
            // =====

            // 3. 불러올 사진 폴더 이름을 인텐트 엑스트에 추가한다.
            // =====
            intent.putExtra("PIC_URL", mSendPicUrl);
            // =====

            // 4. 엑스트라에 1 MB 이상의 데이터 전송
            // =====
            byte[] byte1mb = new byte[1024 * 1024];
            intent.putExtra("DATA_1MB", byte1mb);
            // =====

            // 5. 사진 불러오기 B 액티비티를 실행한다.
            // =====
            startActivityForResult(intent, REQUEST_CODE_PIC_FILE_URL);
            // =====
        }
    }
}
```

액티비티 간 데이터를 주고받을 때의 제약사항 -

생명주기 onPause, onStop, onDestroy에서는 setResult 함수를 사용할 수 없다

244

□ ActivityB.java

@Override

```
protected void onPause() {  
    super.onPause();
```

```
    // 선택된 사진 파일 경로를 전달할 인텐트 생성 및 엑스트라를 설정한다.
```

```
    // =====
```

```
    Intent intent = new Intent();
```

```
    intent.putExtra("RESULT_PIC_FILE_URL", mResultPicFileUrl);
```

```
    // =====
```

```
    // 전달할 인텐트를 설정하고 액티비티를 종료한다.
```

```
    // =====
```

```
    setResult(RESULT_OK, intent);
```

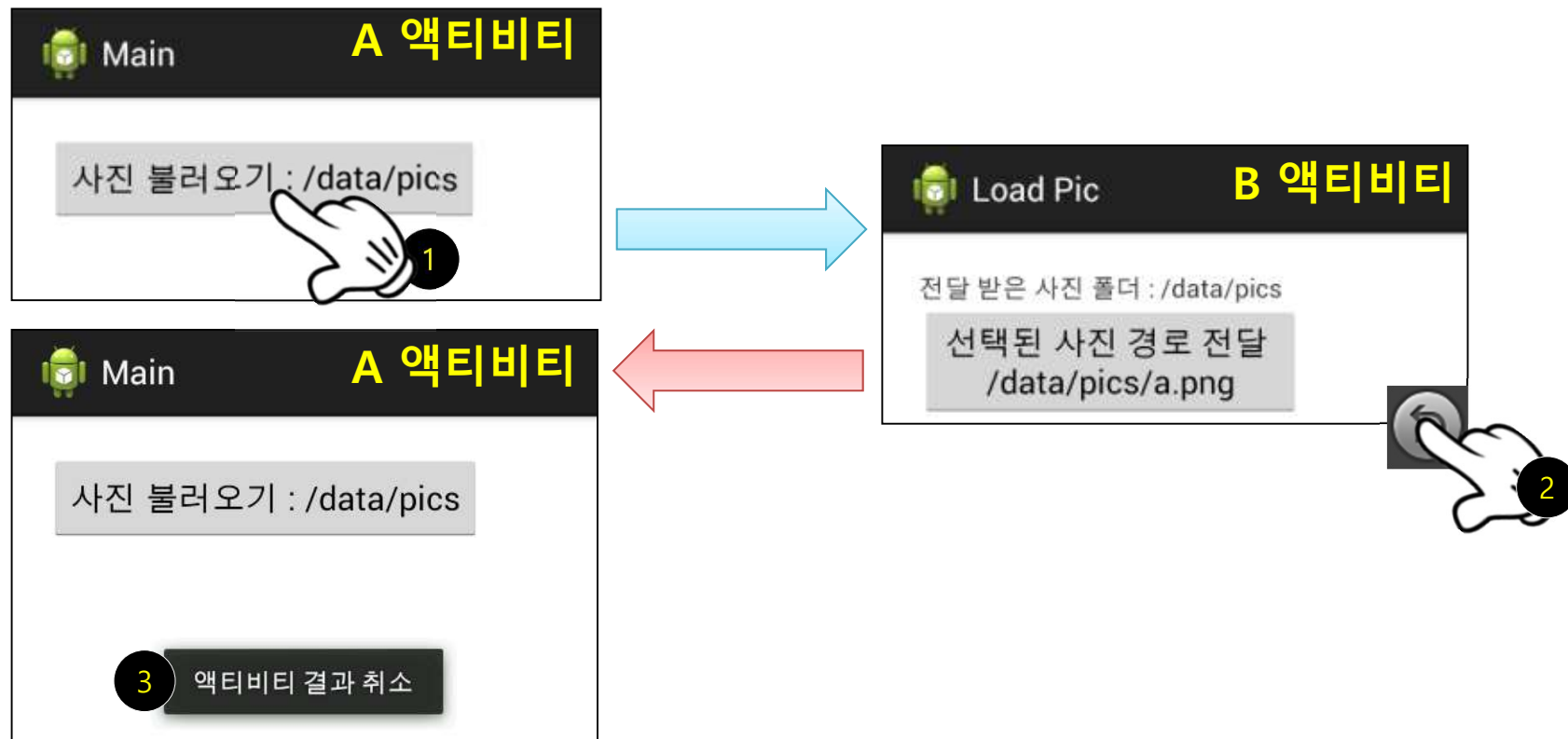
```
    // =====
```

```
}
```

액티비티 간 데이터를 주고받을 때의 제약사항 -

생명주기 onPause, onStop, onDestroy에서는 setResult 함수를 사용할 수 없다

245



액티비티 간 데이터를 주고받을 때의 제약사항 -

생명주기 onPause, onStop, onDestroy에서는 setResult 함수를 사용할 수 없다

프레임워크 소스 경로: framework/base/core/java/android/app/Activity.java

```
public final void setResult(int resultCode, Intent data) {  
    synchronized (this) {  
        mResultCode = resultCode;  
        mResultData = data;  
    }  
}
```

```
...  
public void finish() {  
    if (mParent == null) {  
        int resultCode;  
        Intent resultData;  
        synchronized (this) {  
            resultCode = mResultCode;  
            resultData = mResultData;  
        }  
        if (false) Log.v(TAG, "Finishing self: token=" + mToken);  
        try {  
            if (resultData != null) {  
                resultData.setAllowFds(false);  
            }  
            if (ActivityManagerNative.getDefault()  
                .finishActivity(mToken, resultCode, resultData) )  
            {  
                mFinished = true;  
            }  
        }  
    }  
}
```

finish() → onPause()
onStop()
onDestroy()

이이
Activity Manager에게
결과 전송!

Activity
Manager에게
전달

액티비티 간 데이터를 주고받을 때의 제약사항 -

생명주기 onPause, onStop, onDestroy에서는 setResult 함수를 사용할 수 없다

