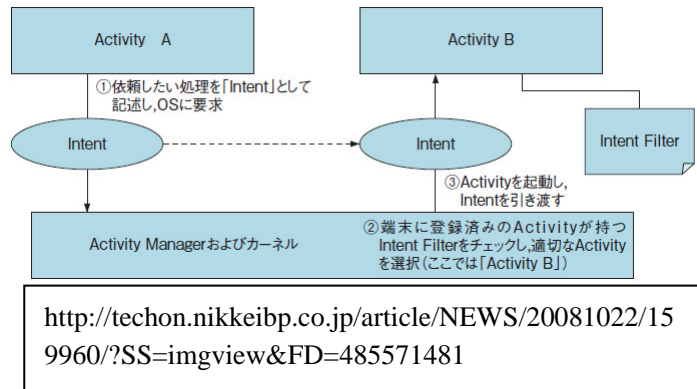


public class **Intent** extends Object implements Parcelable Cloneable

英和：意図，目的

intent が“意図”“目的”を意味するように、Android ではアプリ（アクティビティ）が何をしたいかという「意図・目的」のリクエスト・メッセージをシステムに送ると、システムがそれを解釈・判断し、適切なアクティビティへ渡す仕組みが備わっている。そのために、各アクティビティの AndroidManifest.xml の <intent-filter> には、category, action, data などが宣言されており、これらによりどのような intent に対して応答するかが決まる。



別画面へ遷移させたい、別アプリへ移動し処理をしたい、などアクティビティ間の連携に intent を利用する。

明示的インテント (**Explicit Intents**) と暗黙的インテント (**Implicit Intents**) がある。明示的インテントでは、指定したアクティビティを起動するが、暗黙的インテントでは、明示的に起動させるアクティビティを指定せずに、インテントのパラメータに起動するアクティビティを推測できる情報 (action, data, category) を入れておくと、該当するアクティビティが起動する。

インテントの解説

<http://developer.android.com/reference/android/content/Intent.html>

<http://www.limy.org/program/android/intent.html>

<http://techbooster.jp/andriod/application/294/>

インテントフィルターの解説 <http://developer.android.com/guide/topics/intents/intents-filters.html>

インテントの実例

http://d.hatena.ne.jp/unagi_brandnew/20100309/1268115942

Constructors

Intent() Create an empty intent.

Intent(Context packageContext, Class<?> cls) Create an intent for a specific component.

Intent(String action, Uri uri) Create an intent with a given action and for a given data url.

Intent(String action, Uri uri, Context packageContext, Class<?> cls)

Create an intent for a specific component with a specified action and data.

例) 明示的インテント (自アプリ内のアクティビティを起動)

第1引数は context で検索範囲を指定するが、activity は context を継承しているので、this でも可。

```
Intent i = new Intent( this , xxx.class );
```

```
startActivity( i );
```

```
Intent i = new Intent( this, XXXXX.class );
```

```
    i.setAction( Intent.ACTION_VIEW );           // Activity Action: Display the data to the user.
```

```
startActivity(i);
```

Manifest の application ノードの子ノードの activity ノードに、呼び出す XXXXX を登録する。
マニフェスト・エディタ：アプレケーションノード，追加，Activity，Name (参照)，保存

例) 暗黙的インテント (jpeg を送信するアクティビティへの intent)

```
Intent i = new Intent();
    i.setAction( Intent.ACTION_SEND ); // action
    i.setType( "image/jpeg" );        // MIME type
    Uri uri = Uri.parse( "jpeg ファイルへの path" );
    i.putExtra( Intent.EXTRA_STREAM , uri );
startActivity( i );
    注) path は絶対パス指定  file:///sdcard/maps/12315464.jpeg
```

Intent.EXTRA_STREAM から Uri を取得するためには getParcelableExtra メソッドを使用

```
Intent i = getIntent(); //Intent の取得
Uri uri; //画像の Uri 取得
if ( Intent.ACTION_SEND.equals( i.getAction() ) ) {
    uri = i.getParcelableExtra( Intent.EXTRA_STREAM );
}
}
```

例) ブラウザへ

```
Intent i = new Intent( Intent.ACTION_VIEW , Uri.parse("http://www.google.com") );
startActivity(i);
```

例) Google MapsActivity へ

```
Intent i = new Intent();
String mapurl = "geo:38.230844,140.323316";
i.setAction( Intent.ACTION_VIEW );
i.setClassName( "com.google.android.apps.maps" , "com.google.android.maps.MapActivity" );
i.setData( Uri.parse(mapurl) );
startActivity(i);
```

例) Google MapsActivity で経路探索へ

```
Intent i = new Intent();
i.setAction(Intent.ACTION_VIEW);
i.setClassName( "com.google.android.apps.maps" , "com.google.android.maps.MapActivity" );
// packageName,                className
i.setData( Uri.parse( http://maps.google.com/maps?myl=saddr&daddr=&dirflg=d ) );
startActivity(i);
```

例) 音楽プレーヤーへ

```
Uri uri = Uri.parse( file:///sdcard/song.mp3 );
Intent i = new Intent( Intent.ACTION_VIEW, uri );
i.setType( "audio/mp3" ); // MIME type を設定
startActivity(i);
```

```
Uri uri = Uri.withAppendedPath( MediaStore.Audio.Media.INTERNAL_CONTENT_URI, "1" );
Intent i = new Intent( Intent.ACTION_VIEW , uri );
startActivity(i);
```

例) メールへ

```

Uri uri = Uri.parse( mailto:hoge@hoge.jp );
Intent i = new Intent( Intent.ACTION_SENDTO, uri );
i.putExtra( Intent.EXTRA_SUBJECT, "タイトル" );
i.putExtra( Intent.EXTRA_TEXT, "本文" );
i.setFlags( Intent.FLAG_ACTIVITY_NEW_TASK );
startActivity(i);

Intent i = new Intent(Intent.ACTION_SEND);
i.putExtra(Intent.EXTRA_SUBJECT, "タイトル");
i.putExtra(Intent.EXTRA_STREAM, "file:///sdcard/mysong.mp3");
i.setType("audio/mp3"); // MIME type
startActivity(Intent.createChooser(i, "Choose Email Client"));

```

例) 値を渡したい場合

```

i = new Intent( getApplicationContext(), xxxx.class );
i.putExtra( "KEYWORD", keywordStr );
i.putExtra( "AND", andRadio.isChecked() );
startActivity(i);

```

呼び出された Activity 側では、以下のようにして受け取ります。

```

Intent i = getIntent();
String keyword = i.getStringExtra( "KEYWORD" );
boolean isAnd = i.getBooleanExtra( "AND", true );

```

例) 起動させたアクティビティから戻り値が必要な場合

```

Intent i = new Intent( getApplicationContext() , xxxx.class );
Private final static int REQUEST_CODE = 1;
startActivityForResult( i , REQUEST_CODE );
//第2 引数はリクエストコードで、 どの起動に対する戻り値か判定するのに利用する
protected void onActivityResult( int requestCode , int resultCode , Intent data ) {
    if( requestCode == REQUEST_CODE ){
        if( resultCode == RESULT_OK){
            //戻り値 (Intent data) はこのメソッドで処理する
            String name = data.getStringExtra("NAME");
            . . .
        }
    }
}

```

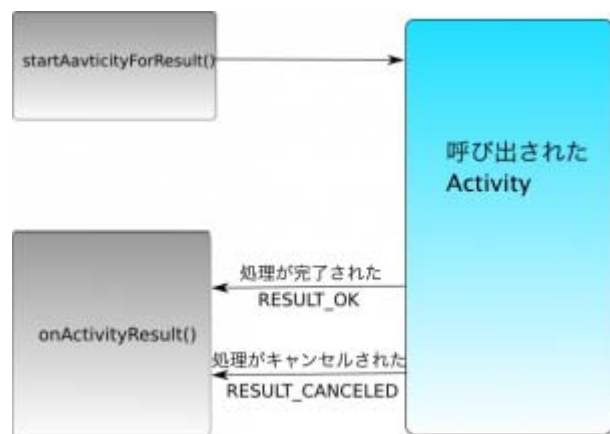
呼び出されたアクティビティ側では以下のよう
に記述してデータを戻す

```

i = new Intent();
//キーと値をセットし、 データを渡す
i.putExtra( "ID", id );
i.putExtra( "NAME", name );
setResult( RESULT_OK , i );
finish();

```

注) Activity.RESULT_OK



android 暗黙 Intent の起動 画像を送る <http://labs.appshelf.info/2011/08/29/453/>

カメラの画像プレビューからのIntentから画像を取得する <http://techbooster.jp.org/android/application/3357/>

言葉の海のプログラマー

Android 標準ブラウザから呼び出されるアプリの作り方

AndroidManifest.xml

```
01     <activity
02         android:name=".MyActivity"
03         android:label="@string/app_name">
04
05         <intent-filter>
06             <category android:name="android.intent.category.DEFAULT" />
07             <data android:mimeType="text/plain" />
08             <action android:name="android.intent.action.SEND" />
09         </intent-filter>
10
11     </activity>

1     if(Intent.ACTION_SEND.equals(getIntent().getAction())){
2         String url = getIntent().getExtras().getCharSequence(Intent.EXTRA_TEXT).toString();
3
4     }
```

Intent で他のアプリと連携するアプリの作り方

ACTION_SEND を指定した Intent を投げる。すると androidManifest.xml で Intent.ACTION_SEND を受け取るように指定されているアプリの一覧が表示され、そのアプリにテキストなどのデータを渡すことができる。

```
1     Intent send = new Intent( Intent.ACTION_SEND );
2     send.setType("text/plain");
3     send.putExtra( Intent.EXTRA_TEXT , "共有したい文字列" );
4     try {
5         startActivity( Intent.createChooser( send , "Select an action
for URL" ) );
6     }catch( android.content.ActivityNotFoundException e ){
7         //該当する Activity がないときの処理
8     }
```

Intent.ACTION_SEND を生成し、そこに putExtra() メソッドで他のアプリに渡したい情報を付加する。Intent.createChooser() の第二引数はアプリ選択ダイアログの上部タイトルになる文字列である。



```

001 package com.gclue.ImageSample;
002
003 import java.io.File;
004
005 import android.app.Activity;
006
007 import android.content.ContentResolver;
008
009 import android.content.Intent;
010
011 import android.graphics.Bitmap;
012
013 import android.net.Uri;
014
015 import android.os.Bundle;
016
017 import android.os.Environment;
018
019 import android.provider.MediaStore;
020
021 import android.view.View;
022
023 import android.view.View.OnClickListener;
024
025 import android.widget.Button;
026
027 import android.widget.ImageView;
028
029 import android.widget.Toast;
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070

```

```

071     }
072     /* startActivityForResult()で Activity が起動し、処理が終わった際に呼ばれる
073     * OutOfMemory が出る 可能性あり
074     */
075     // protected void onActivityResult( int requestCode , int resultCode , Intent data ) {
076     //     if (resultCode == RESULT_OK && requestCode == 1) {
077     //         // MediaStore.EXTRA_OUTPUT で指定した場所と同じ場所から画像を読み込み
078     //         File mFile = new File(Environment.getExternalStorageDirectory(), "test.jpg");
079     //         Uri mUri = Uri.fromFile(mFile);
080     //         // ImageView に渡す
081     //         photoImageView.setImageURI(mUri);
082     //     }
083     //     else if (resultCode == RESULT_OK && requestCode == 2) {
084     //         Uri mUri = data.getData();
085     //         photoImageView.setImageURI(mUri);
086     //     }
087     // }
088     /**
089     * startActivityForResult()で Activity が起動し、処理が終わった際に呼ばれる
090     * (OutOfMemory の修正をしたつもり)
091     */
092     protected void onActivityResult( int requestCode , int resultCode , Intent data ) {
093         if ( resultCode == RESULT_OK && requestCode == 1 ) {
094             // MediaStore.EXTRA_OUTPUT で指定した場所と同じ場所から画像を読み込み
095             File mFile = new File( Environment.getExternalStorageDirectory() , "test.jpg" );
096             Uri mUri = Uri.fromFile( mFile );
097             // 画像を取得
098             ContentResolver conReslv = getContentResolver();
099             if ( mUri != null ) {
100                 try {
101                     // ビットマップ画像を取得
102                     Bitmap bitmap = MediaStore.Images.Media.getBitmap(conReslv, mUri);
103                     // ImageView に渡す
104                     photoImageView.setImageBitmap(bitmap);
105                 } catch (OutOfMemoryError e) {
106                     Toast.makeText(this, "OutOfMemory 発生しました", Toast.LENGTH_SHORT).show();
107                 } catch (Exception e) {
108                     e.printStackTrace();
109                 }
110             }
111         }
112         else if (resultCode == RESULT_OK && requestCode == 2) {
113             // 画像 URI を取得
114             Uri photoUri = data.getData();
115             // 画像を取得
116             ContentResolver conReslv = getContentResolver();
117             if (photoUri != null) {
118                 try {

```

```
122             // ビットマップ画像を取得
123             Bitmap bitmap = MediaStore.Images.Media.getBitmap(conReslv, photoUri);
124             // ImageView に渡す
125             photoImageView.setImageBitmap(bitmap);
126         } catch (OutOfMemoryError e) {
127             Toast.makeText(this, "OutOfMemory 発生しました ",
Toast.LENGTH_SHORT).show();
128         } catch (Exception e) {
129             e.printStackTrace();
130         }
131     }
132 }
133 }
134 }
```