

C-LIS

Android Developers' Gig in KOBE 2011. 8. 27

Intentによるデータ共有

Agenda

- Intentにデータを含めて、別のActivityの呼び出し時にデータを渡す
- ”暗黙的Intent”で、他のアプリと連携する
- 他のアプリに一時的なアクセス権を設定して、セキュアな領域に置いたデータを読み込む

画像表示アプリ

- 画像をダウンロードして、画面に表示する
 - MainActivity - 画像をダウンロード (assetsからコピーする)
 - ViewActivity - 画像を表示

MainActivity



ViewActivity



MainActivity

```
package jp.co.c_lis.sample.securefileshare.ui;

public class MainActivity extends Activity implements OnClickListener {
    private static final String LOG_TAG = "MainActivity";

    private static final String TARGET_FILE_NAME = "IMG_20110420_145845.jpg";

    private File mSaveTo = null;

    /*
     * (non-Javadoc)
     * @see android.app.Activity#onCreate(android.os.Bundle)
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        findViewById(R.id.btn_show_picture).setOnClickListener(this);

        // SDカードへ保存
        downloadToExternal();
    }

    private void downloadToExternal() {
        mSaveTo = new File(Environment.getExternalStorageDirectory(), TARGET_FILE_NAME);
        downloadFile(getAssets(), TARGET_FILE_NAME, mSaveTo);
    }

    private void showPicture() {
        String targetPathName = mSaveTo.getAbsolutePath();

        Intent intent = new Intent(this, ViewerActivity.class);
        intent.putExtra(ViewerActivity.KEY_FILE_NAME, targetPathName);
        startActivity(intent);
    }

    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn_show_picture:
                showPicture();
                break;
        }
    }
}
```

MainActivity

```
/**
 * ファイルをダウンロードして保存.
 *
 * @param am
 * @param fileName
 * @param saveTo
 */
private static void downloadFile(AssetManager am, String fileName, File saveTo) {
    InputStream in = null;
    OutputStream out = null;
    if (!saveTo.exists()) {
        try {
            saveTo.createNewFile();

            in = am.open(fileName);
            out = new FileOutputStream(saveTo);
            byte[] buffer = new byte[512];

            int len = 0;
            while ((len = in.read(buffer)) > -1) {
                out.write(buffer, 0, len);
            }
        } catch (IOException e) {
            Log.d(LOG_TAG, "IOException", e);
        } finally {
            if (in != null) {
                try {
                    in.close();
                } catch (IOException e) {
                    Log.d(LOG_TAG, "IOException", e);
                }
            }
            if (out != null) {
                try {
                    out.close();
                } catch (IOException e) {
                    Log.d(LOG_TAG, "IOException", e);
                }
            }
        }
    }
}
```

ViewActivity

```
package jp.co.c_lis.sample.securefileshare.ui;

import jp.co.c_lis.sample.securefileshare.R;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.widget.ImageView;

public class ViewerActivity extends Activity {

    public static final String KEY_FILE_NAME = "file_name";

    /*
     * (non-Javadoc)
     * @see android.app.Activity#onCreate(android.os.Bundle)
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.viewer);

        // 表示するファイル名
        String fileName = getIntent().getStringExtra(KEY_FILE_NAME);
        if(fileName != null) {

            // ファイルから画像を読み込み
            Bitmap image = BitmapFactory.decodeFile(fileName);

            // 画像を表示
            ImageView imageView = (ImageView) findViewById(R.id.iv_image);
            imageView.setImageBitmap(image);
        }
    }
}
```


Activityの連携 - ViewActivityの呼び出し

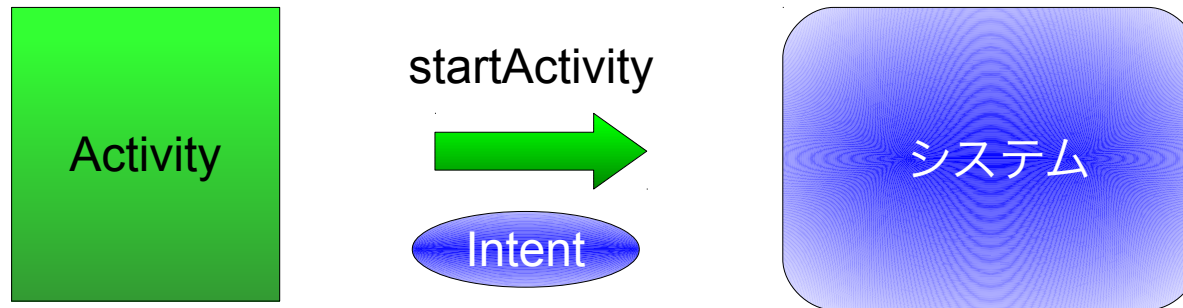
```
private void showPicture() {  
    String targetPathName = mSaveTo.getAbsolutePath();  
  
    Intent intent = new Intent(this, ViewerActivity.class);  
    intent.putExtra(ViewerActivity.KEY_FILE_NAME, targetPathName);  
    startActivity(intent);  
}
```

```
public class ViewerActivity extends Activity {  
  
    public static final String KEY_FILE_NAME = "file_name";  
  
    /*  
     * (non-Javadoc)  
     * @see android.app.Activity#onCreate(android.os.Bundle)  
     */  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.viewer);  
  
        // 表示するファイル名  
        String fileName = getIntent().getStringExtra(KEY_FILE_NAME);
```

Intentとは

- Intent
[名]((形式))1 [U]意図, 意向, (…する) 意志((to do)); 目的; 《法律》意思; 故意.
⇒INTENTION[類語] true intent真意 with intent to ...
(Goo 英和辞書)

システムに「やりたいこと（意図）の内容」を伝える



AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="jp.co.c_lis.sample.securefileshare"
  android:versionCode="1"
  android:versionName="1.0">
  <uses-sdk android:minSdkVersion="4" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <application
    android:icon="@drawable/icon"
    android:label="@string/app_name">
    <activity
      android:name=".ui.MainActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".ui.ViewerActivity"></activity>
  </application>
</manifest>
```

課題

アプリケーションの連携

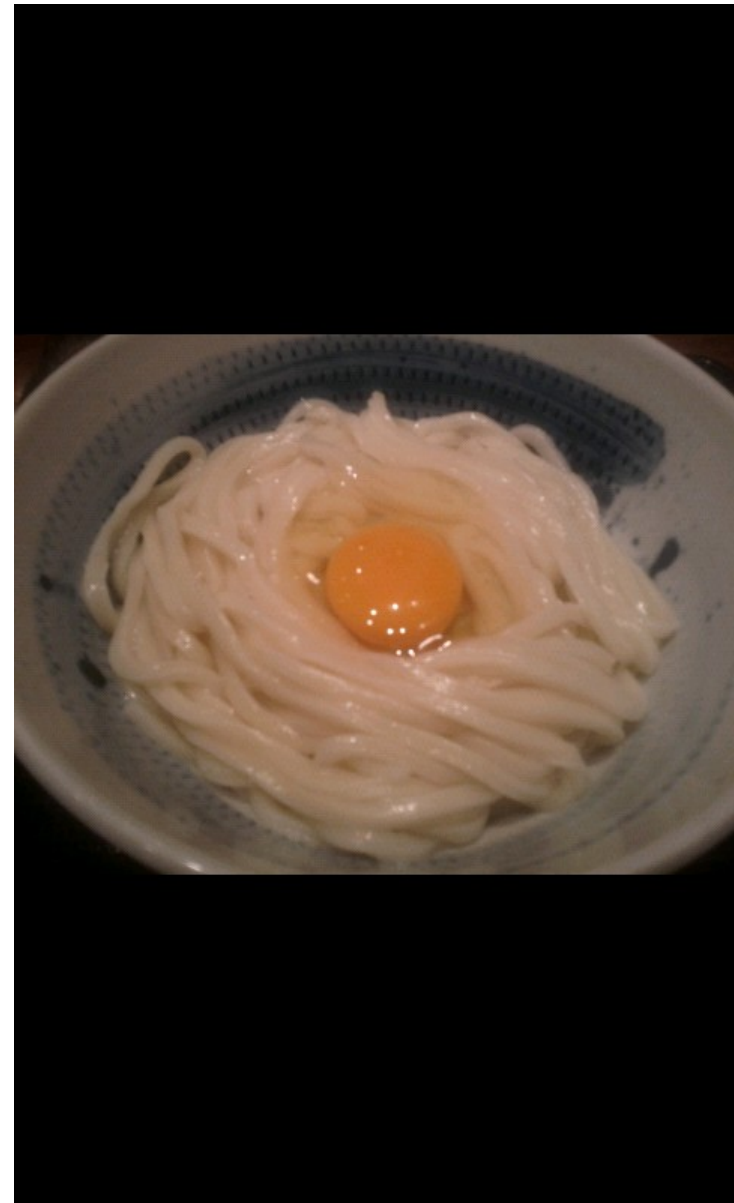
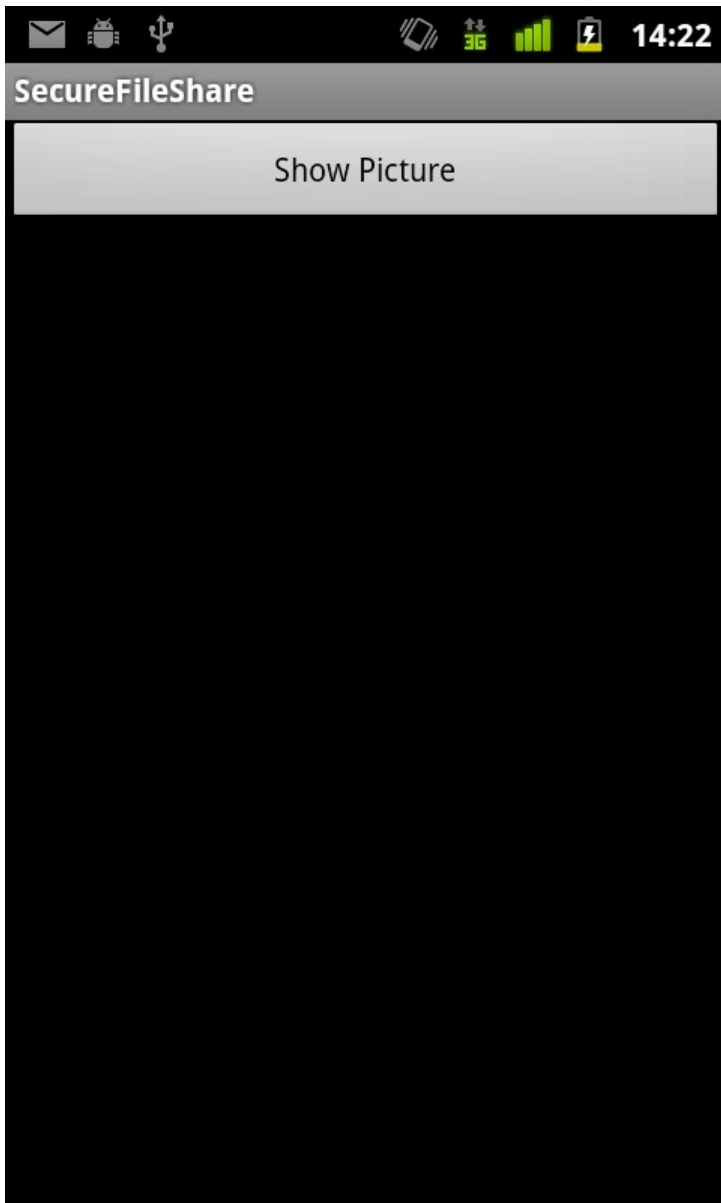
- 画像表示を自分のアプリで実装するのは非効率的
- 他の形式を表示するとき、それぞれに実装する？
 - 音声・動画？
 - PDF？

他のアプリに任せる

暗黙のIntent

```
private void showPictureWithActionView() {  
    String targetPathName = mSaveTo.getAbsolutePath();  
  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setDataAndType(Uri.parse("file://" + targetPathName), "image/jpeg");  
  
    startActivity(intent);  
}
```

```
private void showPicture() {  
    String targetPathName = mSaveTo.getAbsolutePath();  
  
    Intent intent = new Intent(this, ViewerActivity.class);  
    intent.putExtra(ViewerActivity.KEY_FILE_NAME, targetPathName);  
    startActivity(intent);  
}
```



課題

データを守りたい！

- SDカードに保存したファイルは、全てのアプリから読むことが出来る
(セキュアではない)

アプリ内に保存

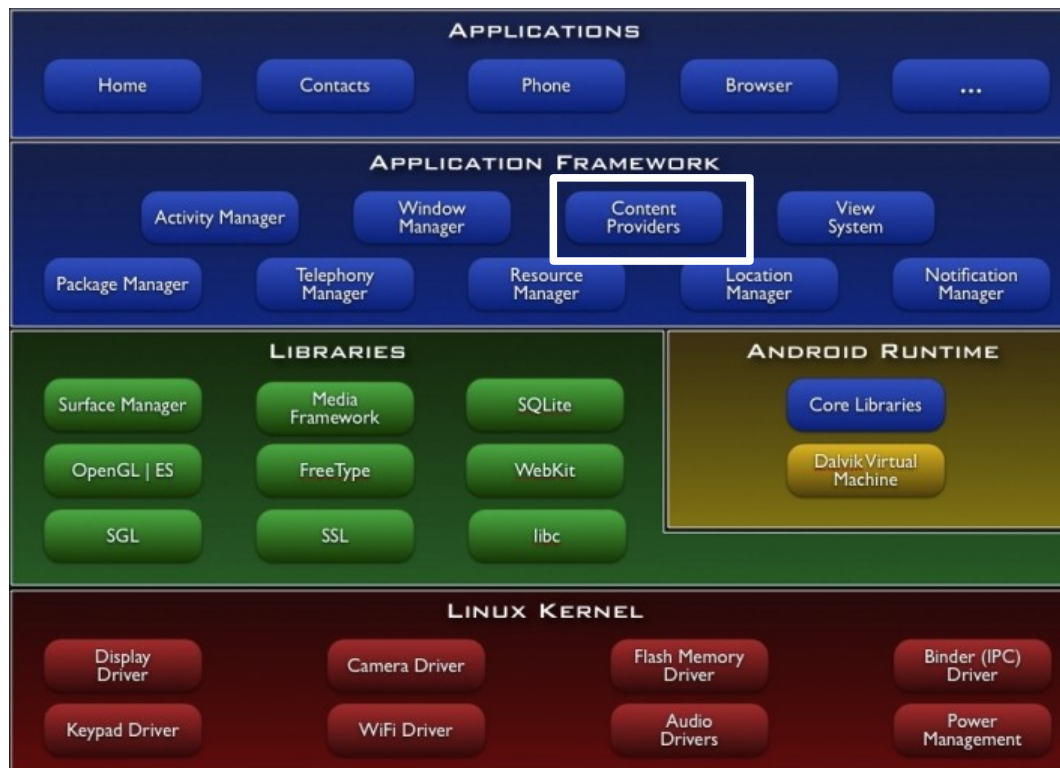
```
private void downloadToExternal() {  
    mSaveTo = new File(Environment.getExternalStorageDirectory(), TARGET_FILE_NAME);  
    downloadFile(getAssets(), TARGET_FILE_NAME, mSaveTo);  
}  
  
private void downloadToInternal() {  
    mSaveTo = new File(getDir("pictures", MODE_PRIVATE), TARGET_FILE_NAME);  
    downloadFile(getAssets(), TARGET_FILE_NAME, mSaveTo);  
}
```

アプリ内に保存したデータは、他のアプリからアクセスが出来ない

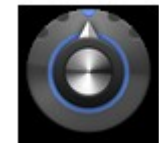
解決方法

ContentProvider

- Androidのシステムコンポーネントの一つ
- 他のアプリケーションとデータを共有するために使う



ContentProviderの例



ContentProviderを追加

```
public class ImageProvider extends ContentProvider {

    private static final boolean DEBUG_FLG = false;
    private static final String LOG_TAG = "ContentProvider";

    public static final String AUTHORITY = "jp.co.c_lis.sample.securefileshare.pictures";

    /*
     * (non-Javadoc)
     * @see android.content.ContentProvider#openFile(android.net.Uri,
     * java.lang.String)
     */
    @Override
    public ParcelFileDescriptor openFile(Uri uri, String mode) throws FileNotFoundException {
        String fileName = uri.getLastPathSegment();
        return ParcelFileDescriptor.open(
            new File(getContext().getDir("pictures", Context.MODE_PRIVATE), fileName),
            ParcelFileDescriptor.MODE_READ_ONLY);
    }

    // 略
}
```

AndroidManifest.xml に記述を追加

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="jp.co.c_lis.sample.securefileshare"
  android:versionCode="1"
  android:versionName="1.0">
  <uses-sdk android:minSdkVersion="4" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <application
    android:icon="@drawable/icon"
    android:label="@string/app_name">
    <activity
      android:name=".ui.MainActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".ui.ViewerActivity"></activity>
    <provider
      android:authorities="jp.co.c_lis.sample.securefileshare.pictures"
      android:grantUriPermissions="true"
      android:name=".provider.ImageProvider">
    </provider>
  </application>
</manifest>
```

呼び出しIntentの変更

```
private void showPictureWithContentProvider() {  
    String targetPathName = mSaveTo.getAbsolutePath();  
  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setDataAndType(  
        Uri.parse("content://" + ImageProvider.AUTHORITY + "/" + targetPathName),  
        "image/jpeg");  
    intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);  
    startActivity(intent);  
}
```

```
public class ImageProvider extends ContentProvider {  
  
    private static final boolean DEBUG_FLG = false;  
    private static final String LOG_TAG = "ContentProvider";  
  
    public static final String AUTHORITY = "jp.co.c_lis.sample.securefileshare.pictures";  
}
```


注意点 (今後の課題)

- 連携先のアプリでセキュアに取り扱われる保証がない
 - 共有操作などでコピーされる可能性がある
- LG Optimusの「ギャラリー」ではエラーが発生
 - 実装に変更（？） ← 調査中

Honeycomb(Android 3.0)以上のGalleryでは、画像(image/*)を ACTION_VIEWで渡すときは、ファイル名に拡張子があると失敗する。

例えば、サンプルコードでは、IMG_20110420_145845.jpgと言う名前で保存していて、これは3.0以上のGalleryでエラーになる。しかし、内部メモリへのコピーの際、拡張子を付けずに保存するとエラーが正常に表示できる。

質疑応答

本資料は、**有限会社シーリス**の著作物であり、
クリエイティブコモンズの表示-非営利-継承 3.0 Unported ライセンスの元で公開しています。



本資料の内容の一部は、**Google**が作成、提供しているコンテンツを複製したもので、
クリエイティブコモンズの表示 2.5 ライセンスに記載の条件に従って使用しています。



ご清聴ありがとうございました。

本日の資料/プロジェクトは、githubで公開します

<https://github.com/keiji>