

1ヶ月で

ANDROIDカメラアプリ開発

@Trunk  
酒本伸也

2016/3/27

# 本日の流れ

- 課題の確認
- Intent
- カメラ呼び出し
- 画像受け取り
- ギャラリー呼び出し
- エディター起動
- 画像シェア
- (暗黙的Intentの条件設定)

# 課題の確認

# 課題

- 自分が作りたいアプリのレイアウトを考え  
実装してみる
- 自分が作りたいアプリのいくつかのwidgetで  
イベントハンドリングを設定する

INTENT



# INTENT

- Activity(Component)同士のやりとりに用いる
- 別のアプリ間でもやりとり可能
- 様々な情報を詰められる(制限あり)

# 2種類ある

- 明示的Intent
  - 呼び出したい相手をComponent名で指定
- 暗黙的Intent
  - 呼び出したい相手を機能から指定

# (唐突)学校に例えると

- 明示的Intent
  - 「佐藤さんちょっと来て」
    - 対象は一人
- 暗黙的Intent
  - 「重い物運べる人ちょっと来て」
    - 誰が来るかわからない
    - 複数人来る場合もある



カメラ呼び出し

# そのまえに

- カメラONにしないとダメ
- 実機ではないのでお忘れなく



# 暗黙的INTENT

- 「カメラ持ってるアプリ来て」

# 「カメラ持ってるアプリ来て」

```
private void launchCamera() {  
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    startActivityForResult(intent, REQUEST_CODE_CAMERA);  
}
```

```
public class MainActivity extends AppCompatActivity {  
    public static final int REQUEST_CODE_CAMERA = 10;  
}
```

- startActivityForResult
  - onActivityResultで  
結果を受け取れる呼び出し方

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode != RESULT_OK) {
        return;
    }
    switch (requestCode) {
        case REQUEST_CODE_CAMERA:
            if (data != null) {
                Bundle bundle = data.getExtras();
                if (bundle.get("data") == null) {
                    return;
                }
                image.setImageBitmap((Bitmap) bundle.get("data"));
            }
            break;
    }
}
```

- resultCode
  - Intentが期待通り終了したかどうか



```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode != RESULT_OK) {
        return;
    }
    switch (requestCode) {
        case REQUEST_CODE_CAMERA:
            if (data != null) {
                Bundle bundle = data.getExtras();
                if (bundle.get("data") == null) {
                    return;
                }
                image.setImageBitmap((Bitmap) bundle.get("data"));
            }
            break;
    }
}
```

- requestCode
  - Intentを送った送信元を判別する数字

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode != RESULT_OK) {
        return;
    }
    switch (requestCode) {
        case REQUEST_CODE_CAMERA:
            if (data != null) {
                Bundle bundle = data.getExtras();
                if (bundle.get("data") == null) {
                    return;
                }
                image.setImageBitmap((Bitmap) bundle.get("data"));
            }
            break;
    }
}
```

- data
  - Intent先での結果が詰められている  
(Nullable)

# 画像受け取り

# BITMAPは大きい

- Bitmapは画像なので容量が大きい

- Intentに詰めるには苦しい

- URIでやりとりする

- [https://ja.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](https://ja.wikipedia.org/wiki/Uniform_Resource_Identifier)



# 「カメラ持ってるアプリ来て」

```
private void launchCamera() {  
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    fileUri = Util.getOutputMediaFileUri(Util.MEDIA_TYPE_IMAGE);  
    intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri); // set the  
    startActivityResult(intent, REQUEST_CODE_CAMERA);  
}
```

```
private Uri fileUri;
```

- 「ただし結果はfileUriに入れてください」
  - Util.getOutputMediaFileUriで  
結果のUriを作成



# URIを生成

- Googleのチュートリアルから拝借
  - <https://developer.android.com/intl/ja/guide/topics/media/camera.html#saving-media>
  - 説明も割愛

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == RESULT_OK) {
        switch (requestCode) {
            case REQUEST_CODE_CAMERA:
                Log.d(TAG, "fileUri: " + fileUri);
                break;
        }
    }
}
```

- fileUriにどんな値が入っているのか確認してみよう
  - 先ほどと異なり dataはnullなので注意。

ギャラリー呼び出し

# 暗黙的INTENT

- 「画像一覧が見たい」

# 「画像一覧が見たい」

```
private void launchGallery() {  
    Intent intent = new Intent(Intent.ACTION_PICK,  
        android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);  
    startActivityForResult(intent, REQUEST_CODE_GALLERY);  
}
```

- 見た上に一つ選択したい



```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == RESULT_OK) {
        switch (requestCode) {
            case REQUEST_CODE_GALLERY:
                Log.d(TAG, "fileUri: " + fileUri);
                fileUri = data.getData();
                launchEditor(fileUri);
                break;
            case REQUEST_CODE_CAMERA:
                Log.d(TAG, "fileUri: " + fileUri);
                launchEditor(fileUri);
                break;
        }
    }
}
```

- data.getData() で  
ギャラリーから選択した画像のUriを取得
- カメラ・ギャラリーどちらにおいてもエディタ一起動のためにUriを渡す

エディター起動

# 明示的INTENT

- SDKのComponentを指定
- 「Aviaryのエディター来て」

# 「AVIARYのエディター来て」

```
private void launchEditor(Uri uri) {  
    Intent imageEditorIntent = new AdobeImageIntent.Builder(this)  
        .setData(uri)  
        .build();  
    startActivityForResult(imageEditorIntent, REQUEST_CODE_EDITOR);  
}
```

- AdobeImageIntent
  - SDKのクラス
- 加工したい画像のUriを渡す



# 中を見ると

- AdobelmageIntent.class

```
public Builder(Context context) {  
    this.intent.putExtra("app-id", context.getPackageName());  
    this.intent.setComponent(new ComponentName(context, AdobeImageEditorActivity.class));  
}
```

- AdobeImageEditorActivity.class
- 特定のComponentを指定している。



```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == RESULT_OK) {
        switch (requestCode) {
            case REQUEST_CODE_EDITOR:
                editedFileUri = data.getData();
                Log.d(TAG, "editedFileUri: " + editedFileUri);
                image.setImageURI(editedFileUri);
                break;
        }
    }
}
```

```
private Uri editedFileUri;
```

- data.getData() で加工した画像のUriを取得
- 取得したUriからImageViewに画像をセット

画像シェア

# 暗黙的INTENT

- 「画像送りたいんだけど誰か」

# 「画像送りたいんだけど誰か」

```
private void shareImage() {  
    Intent shareIntent = new Intent();  
    shareIntent.setAction(Intent.ACTION_SEND);  
    shareIntent.putExtra(Intent.EXTRA_STREAM, editedFileUri);  
    shareIntent.setType("image/jpeg");  
    startActivity(shareIntent);  
}
```

- setAction
  - 送りたい
- putExtra
  - 渡すデータ
- setType
  - データのフォーマット



# これでも同じ

```
private void shareImage() {  
    Intent shareIntent = new Intent(Intent.ACTION_SEND);  
    shareIntent.putExtra(Intent.EXTRA_STREAM, editedFileUri);  
    shareIntent.setType("image/jpeg");  
    startActivity(Intent.createChooser(shareIntent, getString(R.string.share_title)));  
}
```

```
private void shareImage() {  
    Intent shareIntent = new Intent();  
    shareIntent.setAction(Intent.ACTION_SEND);  
    shareIntent.putExtra(Intent.EXTRA_STREAM, editedFileUri);  
    shareIntent.setType("image/jpeg");  
    startActivity(Intent.createChooser(shareIntent, getString(R.string.share_title)));  
}
```

- カメラ呼び出しに似てる



# 恐らくたくさん反応する

- Twitter, Facebook, Instagram, LINE...

- その選択肢が何を意味するのか

タイトルをつけてユーザに分かりやすくする

下に結果が表示されます



ツイートで共有

1回のみ

常時

別のアプリの使用

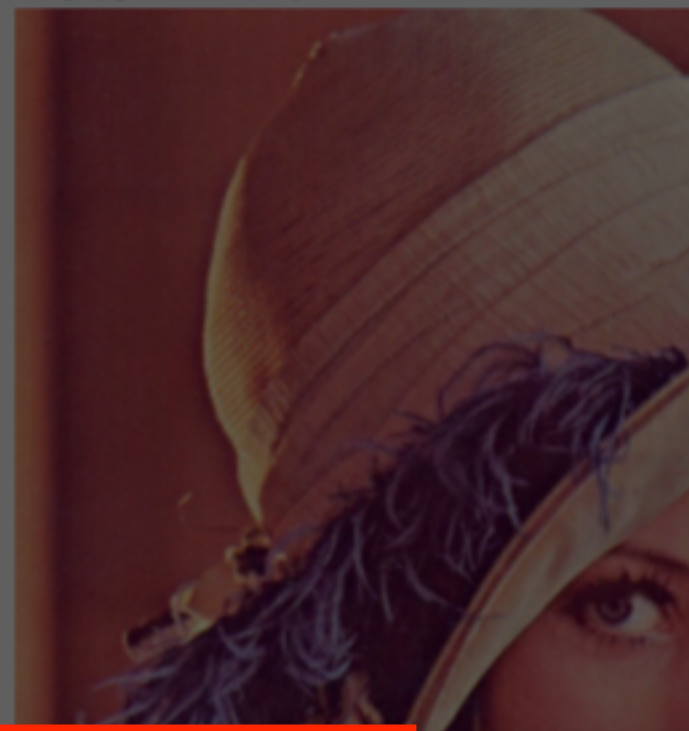


ダイレクトメッセージ



チェックインに追加

下に結果が表示されます



画像をシェアしたい



ダイレクトメッセージ



ツイート



チェックインに追加

```
private void shareImage() {  
    Intent shareIntent = new Intent(Intent.ACTION_SEND);  
    shareIntent.putExtra(Intent.EXTRA_STREAM, editedFileUri);  
    shareIntent.setType("image/jpeg");  
    startActivity(Intent.createChooser(shareIntent, getString(R.string.share_title)));  
}
```

- Intent.createChooserでタイトルをつけられる
- 文字列はxmlで定義して動的に取得する。
  - res/values/strings.xml
  - 基本的にベタ書きはしない

# 暗黙的INTENTの条件設定



# どうして彼らは反応するのか

- 自分たちでも作れるのか？

作れないはずがない。

- しかしこのカメラアプリは

何にも反応しない

- 違いは？



# 反応できることを記述

- AndroidManifest.xmlに  
どのComponentが  
なにに反応するのか定義する。

# 暗黙的INTENTの条件設定

- 条件設定
  - Action
  - Category
  - Data

# ACTION

- どんなことをさせたいのか
  - Intent.ACTION\_VIEW
    - 見る
  - Intent.ACTION\_SEND
    - 渡す

# CATEGORY

- どんなカテゴリのコンポーネントに  
応答してもらうか
  - Intent.CATEGORY\_DEFAULT
    - 暗黙的Intentに対応している
  - Intent.CATEGORY\_LAUNCHER
    - ランチャーに表示している

# DATA

- コンポーネントに渡したいData
- `putExtra(String extra, Bundle data);`
  - `Intent.EXTRA_TEXT`
    - ACTION\_SENDで渡したいData(文字列)
  - `Intent.EXTRA_STREAM`
    - ACTION\_SENDで渡したいData(ファイルパス)



# DATA

- setType(String mimeType)
  - 実ファイルは渡せないなので教える必要がある
    - text/plain
      - テキスト
    - image/jpeg
      - 画像

# 山ほどある

- <https://developer.android.com/intl/ja/reference/android/content/Intent.html>

# 設定例(事例として紹介)

- trunk://~ or http://~ のURLに対して、WebIntentActivityというComponentが反応する
- 実際は受け取った後の処理も必要

```
<activity
  android:name=".activities.WebIntentActivity"
  android:label="WebIntentActivity" >
  <intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:scheme="http"/>
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <category android:name="android.intent.category.BROWSABLE"/>
    <data android:scheme="trunk"/>
  </intent-filter>
</activity>
```