

Android

Programmation

Sylvain Jubertie
sylvain.jubertie@univ-orleans.fr

1 Logs

2 Intents

Organisation du module

8 semaines

- Semaines 1-7 : Cours/TD/TP
- Semaine 8 : Cours/examen écrit/soutenance de projets

1 Logs

2 Intents

Logs

La classe `android.util.Log` permet d'ajouter des messages dans les logs exploitables à partir de la commande `adb logcat`.

Les logs peuvent être de différents types :

```
Log.v( tag , message ); // verbose
Log.d( tag , message ); // debug
Log.i( tag , message ); // information
Log.w( tag , message ); // warning
Log.e( tag , message ); // error
Log.a( tag , message ); // assert
```

L'argument `tag` (généralement le nom de l'activity) est le critère permettant de filtrer les logs.

Exemple

- `Log.e("MyActivity", " File_not_found.");`
- `Log.i("MyActivity", " Connecting_to_...");`

Filtrage

- Uniquement les messages correspondant au tag "MyActivity" :
`adb logcat MyActivity:* *:S`
- Uniquement les erreurs correspondant au tag "MyActivity" :
`adb logcat MyActivity:E *:S`

`*:S` permet de supprimer les messages ne correspondant pas au critère.

1 Logs

2 Intents

Intents

Les *intents* permettent de faire communiquer des *activities*. Par exemple une application de gestion de contacts peut permettre de démarrer l'application de téléphonie pour composer le numéro d'un contact.

Les *intents* sont implantées par la classe `android.content.Intent`. Des informations sur les *activities* susceptibles d'être appelées dans une application doivent être ajoutées dans le fichier `AndroidManifest.xml`.

Exemple: appel à une autre *activity* de l'application

```
public class MainActivity extends Activity
{
    ...
    // Evènements sur un bouton.
    public void startAnotherActivity(View view)
    {
        Log.i(" UserActivity", " button_pressed" );
        Intent intent = new Intent(this, SubActivity.class);
        startActivity( intent );
    }
}
```

Exemple: code de l'*activity* appelée

Code de base de l'autre *activity* :

```
public class SubActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.another);
    }
}
```

Pour l'instant, pas d'échange de données. . .

AndroidManifest.xml correspondent

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android=...
    ...
    <application ...
        <activity android:name="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="SubActivity"
            android:label="@string/subactivity_name"
            android:parentActivityName="MainActivity">
        </activity>
    </application>
</manifest>
```

Passage de données entre *activity*

Il est possible de passer des messages dans l'*intent* par un système de clés-valeurs.

Envoi d'une information :

```
intent.putExtra( String key , ... message );
```

Récupération de l'information :

```
intent.get...Extra( String key );
```

Côté émetteur

```
public class MainActivity extends Activity
{
    ...
    public void startAnotherActivity(View view)
    {
        Intent intent = new Intent(this, SubActivity.class);
        String message = ((EditText)findViewById(R.id.et0))
                        .getText().toString();
        intent.putExtra("text", message);
        startActivity(intent);
    }
}
```

Coté récepteur

```
public class SubActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.another2);
        Intent intent = getIntent();
        String message = intent.getStringExtra("text");
        ((EditText)findViewById(R.id.et0)).setText(message);
    }
}
```

Application Camera

- 1 Lancement de l'application Camera depuis l'appli :
méthode `startActivityForResult`
- 2 Prise de photo avec l'*activity* Camera
- 3 Récupération de l'image dans l'appli :
surcharge de la méthode `onActivityResult`

Bouton de prise de photo

...

```
public void takePhoto(View view) {  
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    startActivityForResult(intent, 0);  
}
```


Surcharge

`@Override`

```
protected void onActivityResult(int requestCode,
                                int resultCode,
                                Intent intent) {

    if (resultCode != 0) {
        Bundle extras = intent.getExtras();
        bitmap = (Bitmap) extras.get("data");
        ImageView image_view =
            (ImageView) findViewById(R.id.image_view);
        image_view.setImageBitmap(bitmap);
    }
    else {
        Toast.makeText(getApplicationContext(), "Cancel.",
                        Toast.LENGTH_SHORT).show();
    }
}
```