Développement Application Android

Expérience Utilisateur

Anthony Chomienne

CPE Lyon

 2023

Développement Application Android — Plan

Service

- Exécute une ou plusieurs tâche en arrière plan
- S'éxecute dans le thread principale de l'application
- Possède son propre cycle de vie
- Peut être automatiquement relancé

AndroidManifest.xml

```
<application ...>
    <service android:name=".MyService"/>
...
</application>
```

Implémentation onStartCommand

- `public int onStartCommand(Intent intent, int flags, int startId);`
- Appeler chaque fois qu'on utilise `startService(Intent);`
- Intent null si le service est relancé automatiquement
- flags vaut 0 ou START_FLAG_RETRY ou START_FLAG_REDELIVERY
- startID: un entier unique identifiant cet appel à onStartCommand();
- doit retourner START_NOT_STICKY ou START_STICKY ou START_REDELIVER_INTENT

Implémentation onBind

- `public IBinder onBind(Intent intent);`

```
public class MyBinder extends Binder
{
    MyService getService() { return
        MyService.this; }
}
```

Implémentation onBind

- `public IBinder onBind(Intent intent);`
- return null ou un objet de type Binder.

```
public class MyBinder extends Binder
{
    MyService getService() { return
        MyService.this; }
}
```

Implémentation onBind

- `public IBinder onBind(Intent intent);`
- return null ou un objet de type Binder.

```
public class MyBinder extends Binder
{
    MyService getService() { return
        MyService.this; }
}
```

- On à accès aux méthode public du service

Implémentation bindService

- `plublic boolean bindService(Intent service, ServiceConnection conn, int flags);`
- Intent représentant le service
- conn listener
- `public void onServiceConnected(ComponentName name, IBinder service);`
- `public void onServicedisconnected(ComponentName name);`
- `unbindService(ServiceConnection conn);`
- flags défini le mode de création du service

Thread

```
Thread thread = new Thread() {
    @Override
    public void run() {
        //do something in background
        //in a loop or not
    }
};
thread.start();
```

Développement Application Android — Plan

Share and Receive

- Permettre de partager des éléments avec d'autres applications
- Permettre de recevoir des éléments d'autres applications
- Augmenter l'intéractivité avec d'autres applications
- Renforcer l'ouverture et la visibilité de votre application

RECEIVE ANDROIDMANIFEST.XML

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category
            android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="image/*" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category
            android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
</activity>
```

Receive Activity

```
Intent receiveIntent = getIntent();
String action = receiveIntent.getAction();
if(Intent.ACTION_SEND.equals(action){
    if(receiveIntent != null &&
       receiveIntent.getType() != null &&
       receiveIntent.getType().contains("image/"))
        Uri uri = receiveIntent.getParcelableExtra(
           Intent.EXTRA_STREAM);
    else if(receiveIntent != null &&
       receiveIntent.getType() != null &&
       receiveIntent.getType().contains("text/"))
        String text = receiveIntent.getStringExtra(
           Intent.EXTRA_TEXT);
}
```

Share Activity

```
Intent sendIntent = new Intent();
sendIntent.setAction(ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "Some text
    to share");
sendIntent.setType("text/plain");
startActivity(sendIntent);
```

Share Activity

```
Uri imageUri;
Intent sendIntent = new Intent();
sendIntent.setAction(ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_STREAM, imageUri);
sendIntent.setType("image/jpeg");
startActivity(sendIntent,
    Intent.createChooser(sendIntent, "Send to");
```

Développement Application Android — Plan

Utilité

- Utilisé pour recevoir des évennements

Utilité

- Utilisé pour recevoir des évennements
- Notifications
- Boot Terminé
- Réseau Disponible
- ...

BroadcastReceiver

```
public MyReceiver extends BroadcastReceiver{
    @Override
    public void onReceive(Context context, Intent
        intent) {
        //do something when receive
    }
}
```

AndroidManifest.xml

```
<application>
    <!-- some activities -->
    <receiver android:name=".BootReceiver" >
        <intent-filter>
            <action android:name=
                "android.intent.action.BOOT_COMPLETED"
                />
        </intent-filter>
    </receiver>
    <receiver android:name=".MyReceiver" >
        <intent-filter>
            <action android:name="myAction">
        </intnet-filter>
    </receiver>
</application>
```

Send a Broadcast Message

```
Intent intent = new Intent();
intent.setAction("myAction");
intent.putExtra("someData", "Some Value");
context.sendBroadcast(intent);
```

Développement Application Android — Plan

Notifications

- Informer l'utilisateur
- Ne dois pas être trop souvent
- Ne pas multiplier inutillement les notifications

```
NotificationManager manager = (NotificationManager)
    context.getSystemService(
    Context.NOTIFICATION_SERVICE);
Notification.Builder builder = new
    Notification.Builder(context);
Notification.InboxStyle style = new
    Notification.InboxStyle();
```

Notification Style

```
style.addLine("A Line");
style.addLine("Another Line");
style.setSummaryText("My App");
style.setBigContentTitle("Notification Title");
builder.setStyle(style);
```

Notifiation Builder

```
builder.setContentText("Notification Title");
builder.setContentTile("My App");
builder.setSmallIcon(R.drawable.logo);
builder.setAutoCancel(true);
```

Start Intent on Notification

```
int id = 12345
Intent newIntent = new
    Intent(context,MainActivity.class);
newIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
PendingIntent pendingIntent =
    PendingIntent.getActivity(context,0,newIntent,
    FLAG_ONE_SHOT);
builder.setContentIntent(pendingIntent);
manager.cancel(id);
manager.notify(id, builder.build());
```

Développement Application Android — Plan

Sources et Biblio

- http://www.idc.com/prodserv/smartphone-os-market-share.jsp
- Part de marché Mobile en France Janvier 2015
- http://opensignal.com/reports/2015/08/android-fragmentation/
- http://developer.android.com/
- http://www.statista.com/
- https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad