

Développement Application Android

Expérience Utilisateur

Anthony Chomienne

CPE Lyon



2020

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 SERVICE
- 2 SHARE & RECEIVE
- 3 BROADCASTRECEIVER
- 4 NOTIFICATIONS
- 5 CONCLUSION

- Exécute une ou plusieurs tâche en arrière plan

- Exécute une ou plusieurs tâche en arrière plan
- S'exécute dans le thread principale de l'application

- Exécute une ou plusieurs tâche en arrière plan
- S'exécute dans le thread principale de l'application
- Possède son propre cycle de vie

- Exécute une ou plusieurs tâche en arrière plan
- S'exécute 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);`

IMPLÉMENTATION ONSTARTCOMMAND

- `public int onStartCommand(Intent intent, int flags, int startId);`
- Appeler chaque fois qu'on utilise `startService(Intent);`

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

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`

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();`

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

- `public boolean bindService(Intent service, ServiceConnection conn, int flags);`

IMPLÉMENTATION BINDSERVICE

- `public boolean bindService(Intent service, ServiceConnection conn, int flags);`
- Intent représentant le service

IMPLÉMENTATION BINDSERVICE

- `public boolean bindService(Intent service, ServiceConnection conn, int flags);`
- Intent représentant le service
- conn listener

IMPLÉMENTATION BINDSERVICE

- `public boolean bindService(Intent service, ServiceConnection conn, int flags);`
- Intent représentant le service
- conn listener
- `public void onServiceConnected(ComponentName name, IBinder service);`

IMPLÉMENTATION BINDSERVICE

- `public 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);`

IMPLÉMENTATION BINDSERVICE

- `public 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);`

IMPLÉMENTATION BINDSERVICE

- `public 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

- 1 SERVICE
- 2 SHARE & RECEIVE
- 3 BROADCASTRECEIVER
- 4 NOTIFICATIONS
- 5 CONCLUSION

SHARE AND RECEIVE

- Permettre de partager des éléments avec d'autres applications

SHARE AND RECEIVE

- Permettre de partager des éléments avec d'autres applications
- Permettre de recevoir des éléments d'autres applications

SHARE AND RECEIVE

- Permettre de partager des éléments avec d'autres applications
- Permettre de recevoir des éléments d'autres applications
- Augmenter l'interactivité avec d'autres applications

SHARE AND RECEIVE

- Permettre de partager des éléments avec d'autres applications
- Permettre de recevoir des éléments d'autres applications
- Augmenter l'interactivité 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

- 1 SERVICE
- 2 SHARE & RECEIVE
- 3 BROADCASTRECEIVER**
- 4 NOTIFICATIONS
- 5 CONCLUSION

- Utilisé pour recevoir des événements

- Utilisé pour recevoir des événements
- Notifications

- Utilisé pour recevoir des événements
- Notifications
- Boot Terminé

- Utilisé pour recevoir des événements
- Notifications
- Boot Terminé
- Réseau Disponible

- Utilisé pour recevoir des événements
- 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

- 1 SERVICE
- 2 SHARE & RECEIVE
- 3 BROADCASTRECEIVER
- 4 NOTIFICATIONS**
- 5 CONCLUSION

NOTIFICATIONS

- Informer l'utilisateur

NOTIFICATIONS

- Informer l'utilisateur
- Ne dois pas être trop souvent

NOTIFICATIONS

- Informer l'utilisateur
- Ne doit pas être trop souvent
- Ne pas multiplier inutilement 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.setContentTitle("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 intent =
    PendingIntent.getActivity(context, 0, newIntent,
        FLAG_ONE_SHOT);
builder.setContentIntent(newIntent);
manager.cancel(id);
manager.notify(id, builder.build());
```

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 SERVICE
- 2 SHARE & RECEIVE
- 3 BROADCASTRECEIVER
- 4 NOTIFICATIONS
- 5 CONCLUSION

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>

Licence : 