



# 行動裝置程式設計

## Unit 07 服務 Service

蘇維宗 (Wei-Tsung Su)  
suwt@au.edu.tw  
564D





# 課程目標





# 應用程式元件(App Components)

每個應用程式 (App) 可以包含多個以下四種應用程式元件。

- **Activities**
  - 對應一個可視的圖形化使用者介面 (GUI)
- **Services**
  - **沒有對應的GUI, 而是在背景執行的程式**
- **Broadcast receivers**
  - 接收與回應系統 (或其它程式) 的廣播訊息
- **Content providers**
  - 提供應用程式間資料的傳遞





## Service (服務)

Service是可以在背景長時間執行的Android應用程式元件。

常見的例子，例如

- 背景撥放音樂
- 背景下載資料
- ...

其它應用程式元件可透過行程間通訊(Inter-Process Communication、IPC)的方式與Service溝通。





# Service的兩種形式

## Unbounded Service

- 呼叫者(caller)透過呼叫 `startService()` 來啟動service
- 呼叫者結束後, service會繼續執行
- Unbounded service通常只執行一項不會回傳結果的工作, 當工作完成後service會自己停止

## Bounded Service

- 呼叫者(caller)透過呼叫 `bindService()` 來連結service
- 同時可以有多個呼叫者連結到同一個service
- 當所有呼叫者切斷與此service的連結後, service會自動結束執行。



# Service的結束方式

## Unbounded Service

- Service結束任務後自己呼叫  
`stopSelf()` 結束
  - 例如, 當音樂播完後 service呼叫  
`stopSelf()` 結束
- 透過其它程式元件呼叫  
`stopService()` 結束所指定的 service
  - 例如, 當按下結束播放後呼叫  
`stopService()` 結束

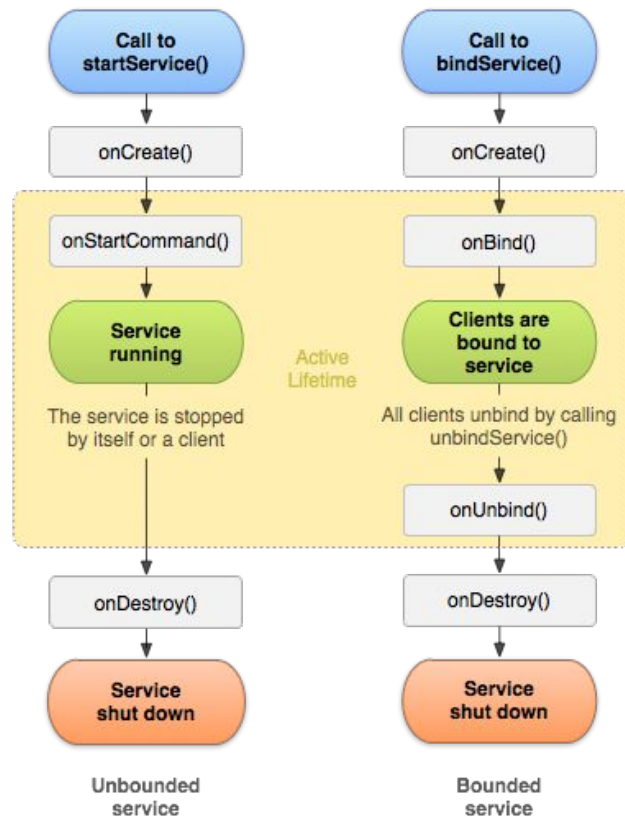
## Bounded Service

- 呼叫者可以呼叫  
`unbundService()` 來切斷與 service的連結。當所有連結到此 service的呼叫者都切斷連結後, 系統會自行結束此 service

## Service的生命週期

預設，service會與宣告此service的應用程式屬於同一個程序(process)，所以當使用者正在與此應用程式互動時，service可能會減緩執行速度以避免使用者認為應用程式當機。為了避免service因為上述原因減緩執行速度而造成效能問題，建議可以在service中產生新的執行緒(thread)。

資料來源: Android Developers



Service生命週期(來源:Android Developers)



# Service類別

## Service

基礎類別，**必須自己實作多執行緒**，所以可以用來實現能夠接受多個要求的服務。

## IntentService

延伸類別，因為一次只處理一個要求，所以開發者只需要實作

`onHandleIntent()`

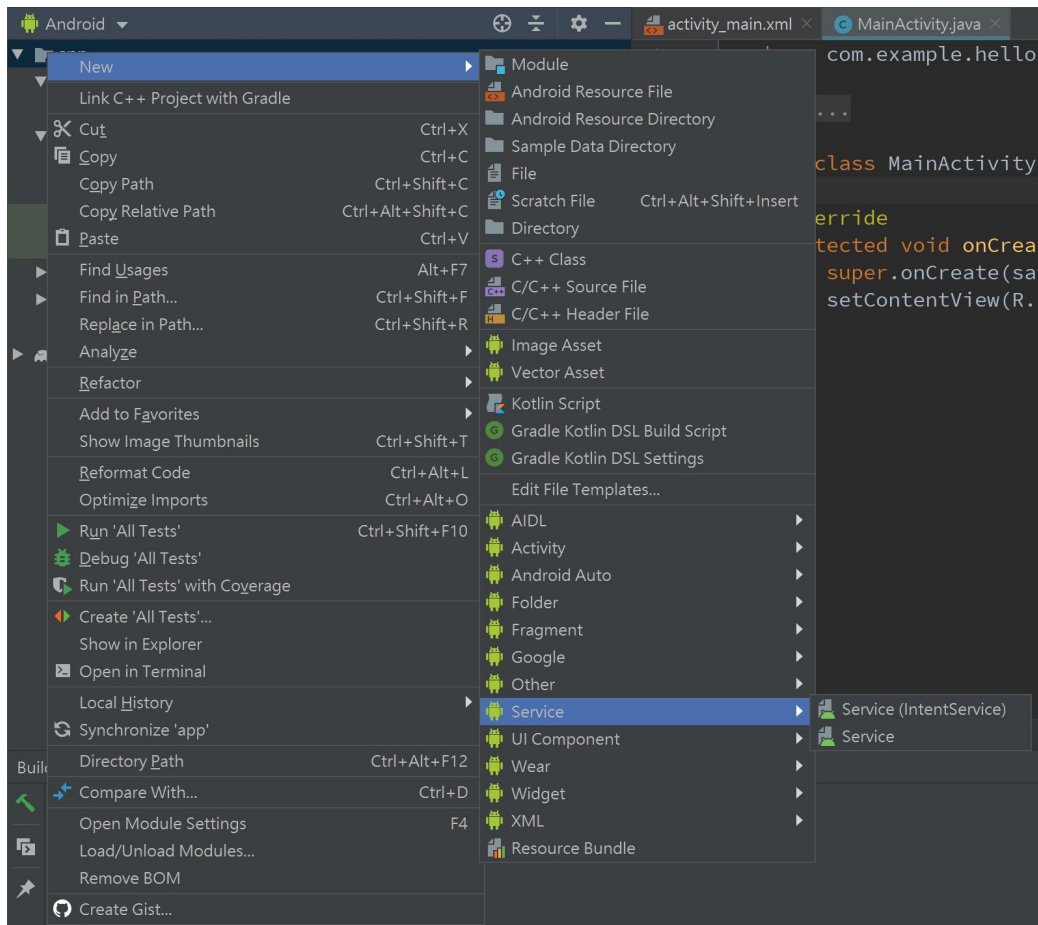
方法來處理該要求即可。



# 新增Service

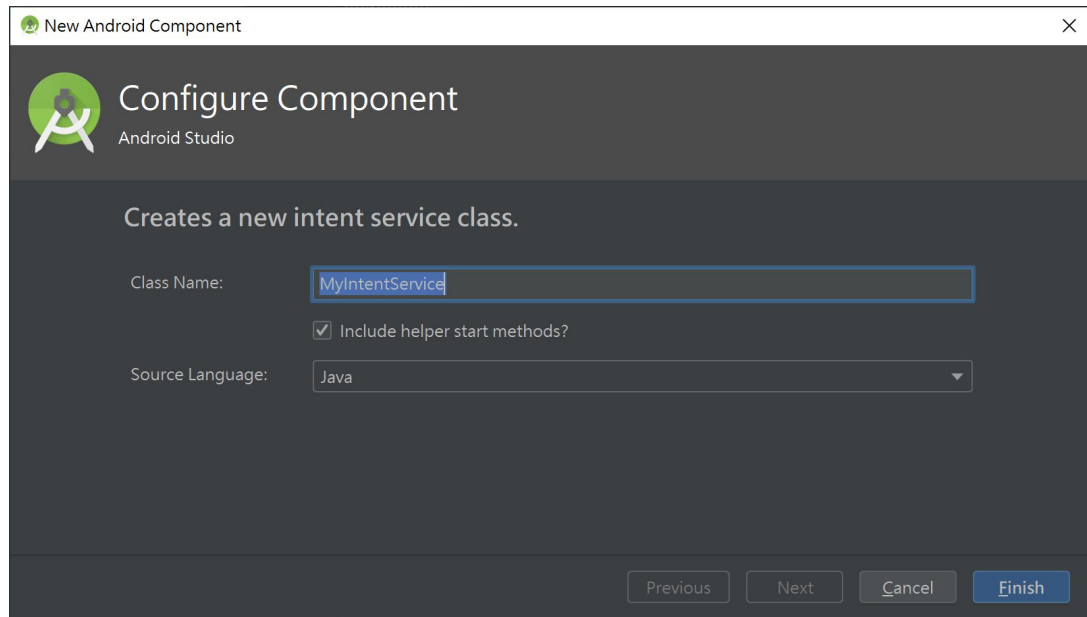
在專案上按滑鼠右鍵並選擇  
新增Service

- 如果service必須處理多個要求，選擇新增Service
- 如果service只會處理一個要求，選擇新增IntentService



## 新增Service (續)

以選擇IntentService為例，  
需要輸入類別名稱(Class  
Name)與使用的程式語言。





## 新增Service後，專案的變化

- 在manifests/AndroidManifest.xml中新增了一個標籤  
`<service`  
    `android:name=".MyIntentService"`  
    `android:exported="false"></service>`
- 在src目錄中新增了一個java原始檔  
src/[package]/**MyIntentService.java**

註: 因為Service沒有使用這介面, 所以不會產生對應的layout檔案。





# 啟動Unbounded Service

Caller: MainActivity.java

```
1. Intent intent = new Intent();  
2. intent.setClass(MainActivity.this, MyIntentService.class);  
3. startService(intent);
```

Callee: MyIntentService.java

```
1. protected void onHandleIntent(Intent intent) {  
2.     // 要Service在背景中(以執行緒)執行的工作  
3. }
```





## 停止Unbounded Service

**方法一:** 在Caller中(即MainActivity.java)呼叫`stopService()`

```
1. stopService(intent);
```

**方法二:** 在Callee中(即MyIntentService.java)呼叫`stopSelf()`

```
1. stopSelf();
```



# Q & A

---



Computer History Museum, Mt. View, CA

