

Android Services

למה נועדו Services ?

services נועדו לפעולות ארוכות טווח

מה ההגדרה של ארוכות טווח?

ארוכות טווח - טווח של זמן בו יש סיכוי שהיוסר יעבור
לאפליקציה אחרת.

Thread ופועולות ארוכות טווח

- thread לבדו לא יכול לטפל בפעולות ארוכות טווח.
- ה- Thread שמופעל בתוך Activity process, פועל כל עוד ה- process Activivty חי.
- כאשר היוסר עובר לאפליקציה אחרת יש סיכוי שה- Activity לא יהיה קיים.

Service

- רכיב - יישום שרץ ברקע ומיועד לבצע פעולות ארוכות
- service מרחיב את המחלקה הקיימת service
- ה-service רץ ב-thread הראשי

יש שתי קטגוריות ל-service

1. startService

- a. יש לו תוכלת חיים משל עצמו.
- b. מוגבל ביכולת שלו לבוא באינטרקציה עם components
- c. רב ה-services הם started

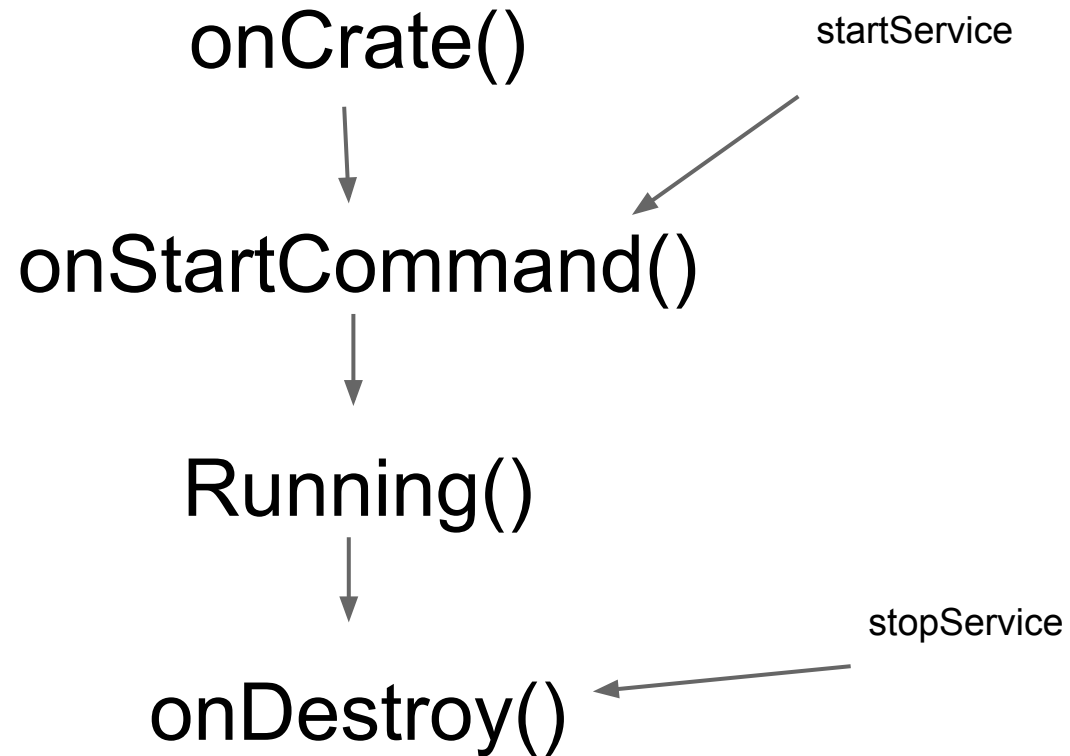
2. bound

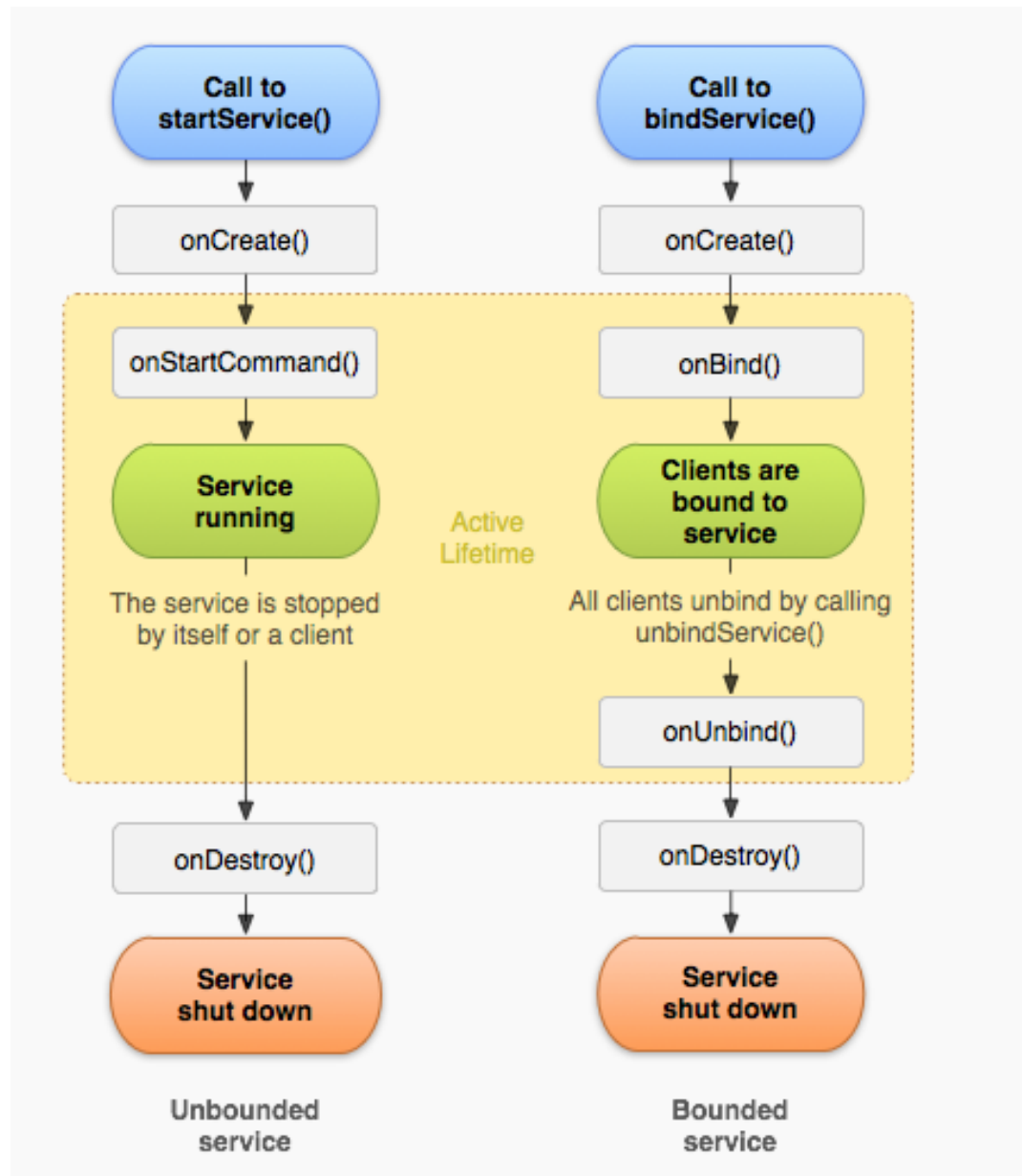
- a. תוכלת החיים תלויה ברכיב שיצר אותו
- b. יש לו אינטרקציה עם הרכיב שיצר אותו
- c. חי כל עוד הרכיב שיצר אותו חי

started()

1. השרות יתחיל כאשר תיקרא הפונקציה startService
2. כאשר ה-service מתחיל הוא יכול לפעול ללא הגבלה
3. כאשר ה-service מסתיים הוא עוצר את עצמו.

מעגל החיים של service started





Service life cycle

ירושם מ- Service

דריסה של onCreate

דריסה של onStartCommand

דריסה של onDestroy

דריסה של onBind - רק אם יצרתם bind services

לא לשכח להוסיף ל-manifest

Providing user feedback

הצג Toast
שימוש ב- Handler

IntentService

תת מחלקה של service המשתמשת ב-Thread נפרד.
צריך לממש את הפונקציה `onHandleIntent()`
המקבלת intent לכל בקשה על מנת שניתן יהיה לעשות
את העבודה ברקע.

- יתרון של IntentService - יוצר בעצמו Thread. ה-IntentService רץ על thread משל עצמו.
- נוצר תור עבודה שמעביר intent אחד בכל פעם למימוש של `onHandleIntent`. אין צורך לדאוג ל-multi thread
- עצירת IntentService על ידי `selfStop`
- `startService` - רץ על ה-Thread הראשי.

יצירת IntentService מבוסס service

- יצירת class שיורש מ - IntentService
- ב- constructor יש לתת שם ל - thread
- דריסה של הפונקציה .IntentService.
onHandleIntent
- רישום ה-service ב- manifest

שרותי מערכת אנדרואיד

Wi-Fi Service

Audio Service

Vibrator Service

Sensor Service

Power Service

Status Bar Notification

Notification manager

- notification manager הוא מנהל ההודעות.
- אובייקט במערכת האנדרואיד שתפקידו לנהל את ההודעות.

-Notification manager

שלבים ליצירה

1. צריך הפניה ל-notification manager.ה-ns הוא service של המערכת.
2. בניית הודעה על ידי הפונקציה - Notification.
3. תגובה לאירוע ההודעה - setLatestEvent
4. שליחת הודעה למנהל ההודעות - notify

PendingIntent

מסר שניתן ליישום אחר לביצוע קטע קוד שמוגדר מראש.

ניתן ליצור PendingIntent במספר דרכים

`getActivity(Context i,int,Intent,int)`

`getBroadcast(Context,int,Intent,int)`

`getService(Context,int,Intent,int)`

העלמת ההודעה

```
public void cancel(int id)
```

```
public void cancelAll()
```

Notification Manager

Must

Icon

Title

pendingIntent

optional :

Ticket text

sound

vibrate

flashing led

הוספת קול להודעה

```
notification.defaults|=Notification.DEFAULT_SOUND
```

העברת מיקום קובץ

```
notification.sound=Uri.parse(file:///sdcard/path/mysound.mp3);
```

העברת מיקום משאב בפרויקט

```
notification.sound=Uri.parse("android.resourc://" + getPackageName + "/" + R.  
raw.s1);
```

הקול יופעל עד שההודעה תטופל :

```
notification.flag|=Notification.FLAG_INSISTENT;
```

הוספת רטט

```
notification.defaults|=Notification.DEFAULT_VIBRATE
```

העברת מיקום קובץ

```
notification.vibrate={0,100,200,400};// array of long
```

צריך להוסיף במינפסט הרשאה ל- :vibrate

```
<user-permission android:name="android.permission.VIBRATE"/>
```

דגלים נוספי

FLAG_NO_CLEAR - הודעה שלא מתבטלת

FLAG_ONGOING_EVENT - הודעה במהלך ארוע

FLAG_AUTO_CANCEL - הודעה שתתבטל שיגעו בה

FLAG_INSISTENT - קול שיופיע עד שההודעה תטופל

FLAG_FOREGROUND_SERVICE - הודעה משרות

Foreground Services

1. שרות חזית מספק הודעה אשר ממוקמת תחת
OnGoing
2. כדי להפעיל את ה-Service צריך לקרוא ל-
startForeground()
3. הפונקציה startForeground מקבלת שני פרמטרים
מספר מזהה, ואת ההודעה שבשורת המצב.
4. על מנת לבטל foreground קוראים לפונקציה
stopForeground(true)

Broadcast reciever

Broadcast reciever

1. מאזין למסרי intent שמשודרים
2. רכיב שמופעל רק כאשר ארועים מסוימים שמתרחשים כמו sms telephone וכו
3. יצירת broadcast reciever
 - a. צריך לרשת מ-BroadcastReciver
 - b. רישום מקלט השידור לארוע (ניתן באמצעות java)
 - c. טיפול בארוע - onRecive
 - d. הריסת broadcast reciever

ראה דמו

יום ללא תכנות הוא יום מבוזבז!