השתלמות Android
Broadcasts

אלון חימוביץ

Broadcast

- הודעת מערכת או הודעה שנשלחת על ידי אפליקציה כשארוע מסויים קורה
  - הודעה לא ממוענת לאפליקציה ספציפית
- אפליקציה יכולה לשלוח הודעות Broadcast
- אפליקציה יכולה להרשם לקבלת הודעות באמצעות- Broadcast Receiver

- הודעות מועברות באמצעות Implicit Intent לאפליקציות שנרשמו לארוע מסויים- עם ACTION שמתאר את הארוע לדוגמה:
- ACTION_BOOT_COMPLETED
- ACTION_POWER_CONNECTED

הערה: בדומה ל pubsub pattern

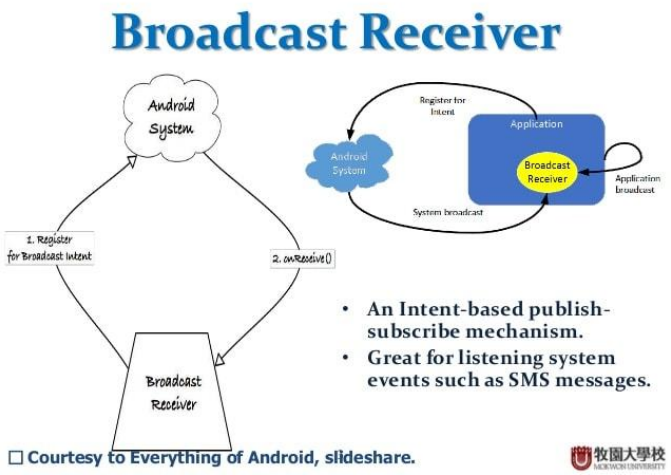
- ניתן גם לשלוח הודעות לאפליקציות ספציפיות באמצעות Explicit Intent (הרחבה...)

SMS_RECEIVED_ACTION

סוגי הודעות Broadcast

- System Broadcasts
  - הודעות שנשלחות על ידי מערכת אנדרואיד כשקורה ארוע שעלוי להשפיע על האפליקציה
- Custom Broadcasts
  - נשלחות על ידי האפליקציה, מודיעות על ארוע שקרה באפליקציה
  - Ordered Broadcast
  - Normal Broadcast - הנפוץ ביותר

Broadcast Receiver



- רכיב באפליקציה שמקבל הודעות Broadcast אליהן הוא נרשם
- אורך החיים שלו קצר, רשאי לבצע מספר פעולות ולסיים
- מופיע כרכיב במערכת בקובץ המניפסט

```
<receiver
    android:name=".PhoneReceiver"
    android:enabled="true"
    android:exported="true" />
```

- ניתן לרשום את הרכיב לסוג הודעה ספציפי בשתי דרכים:
- סטטי - רישום בקובץ ה Manifest - לא משויך ל Activity אלא ברמת אפליקציה.
  - כמות ההודעות שניתן לקבל בדרך זו מצומצמת ומתעדכנת כל גרסה.
- דינאמי - רישום מתוך ה Activity - משויך ל Activity ממנו נרשם.
  - הודעות מערכת (system) או הודעות מוגדרות של האפליקציה (custom)

Broadcast Receiver - סטטי

- הודעת Broadcast מועברת בהתאם למסנן שהוגדר
- Broadcast סטטי - רושמים את המסנן לאחת מהודעות המערכת בקובץ ה Manifest,
- המשמעות היא שההודעה תתקבל גם כשהאפליקציה לא במצב ריצה
- דוגמאות להודעות חשובות שניתן להגדיר סטאטיות:
- ACTION_BOOT_COMPLETED, SMS_RECEIVED_ACTION
- ACTION_EVENT_REMINDER, ACTION_PHONE_STATE_CHANGED

- **<u>חשוב</u>** - להורדת עומס וכמות הודעות שנשלחות לאפליקציות החל מאנדרואיד 8.0 יש מספר מצומצם של הודעות מערכת אליהן ניתן להרשם באופן סטטי.
- רשימה מלאה:
- https://developer.android.com/guide/components/broadcast-exceptions

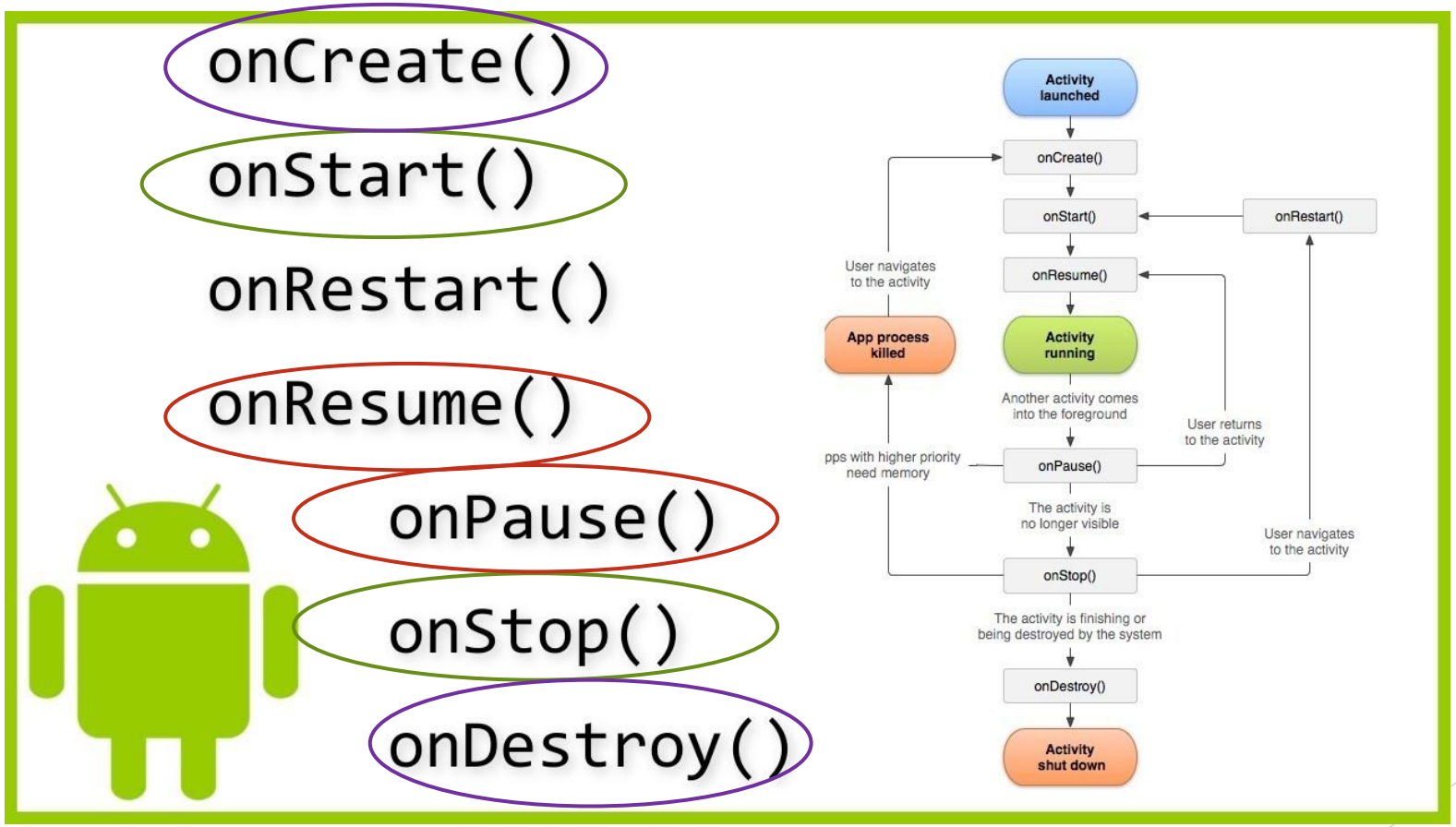
SMS_RECEIVED_ACTION

Broadcast Receiver - דינאמי

- Broadcast Receiver נוצר ונרשם מתוך האפליקציה בזמן הריצה
- המסנן-filter - מוגדר בתכנה בעת יצירת ה Broadcast Receiver ולא מוגדר בקובץ המניפסט
- מקבל הודעות Broadcast כל עוד ה Activity שרשם אותו נמצא במצב ריצה
- חובה לבטל את הרישום לארוע בסיום

SMS_RECEIVED_ACTION

Register/Unregister Dynamic Broadcast



Broadcast Receiver דינאמי להודעת מערכת שלב ראשון -יצירת הרכיב

- צרו אפליקציה חדשה או הוסיפו לאפליקציה קיימת
- נוסיף רכיב Broadcast Receiver לקליטת חיבור של אזניות למכשיר
- הוסיפו את הרכיב באופן הבא:

  New->Other->Broadcast Receiver
- קראו לרכיב בשם HeadsetReceiver
- שימו לב שהשדות Exported,Enabled מסומנים ב V.
- Exported- מאפשר לקבל הודעות Broadcast שנשלחות מחוץ לאפליקציה - קריטי כאשר מעוניינים לקבל הודעות מערכת

Broadcast Receiver דינאמי להודעת מערכת שלב ראשון -יצירת הרכיב

- פעולת onReceive שנוצרה היא הפעולה שתזומן בעת קבלת ה broadcast
- מחקו את החריגה בפעולה - throw new UnsupportedOperationException("Not yet implemented");
- הציגו Toast על המסך על ידי שימוש ב context שהתקבל בפעולה,באופן זה נדע שהתקבל ה broadcast

```
public class HeadSetReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
         Toast.makeText(context,“Broadcast received",Toast.LENGTH_SHORT).show();


}
```

Broadcast Receiver דינאמי להודעת מערכת שלב שני -רישום הרכיב

- ברכיב שלנו מעוניינים לקבל הודעה כאשר יש ארוע של חיבור של אזניות
- ה Broadcast נשלח עם implicit intent כאשר שדה ה Action מציין את הארוע
- בעבור חיבור או ניתוק אזניות ה Action הוא : android.intent.action.HEADSET_PLUG
- ל intent יש שדות נוספים שמציינים האם מדובר בחיבור או ניתוק של האזניות, האם מדובר באזניות עם מיקרופון מובנה ועוד...
- יצירת מופע ורישום של Broadcast דינאמי - באחריות ה Activity שבו מעוניינים לקבל את ההודעה.

Broadcast Receiver דינאמי להודעת מערכת שלב שני -רישום הרכיב

- ב Activity ניצור מופע של הרכיב ונרשום אותו באופן הבא:
- נגדיר הפנייה כתכונה -

```
private HeadSetReceiver receiver;
```

- בפעולת onCreate ניצור מופע:

```
// create receiver instance
receiver= new HeadSetReceiver();
```

- לטובת הגדרת ה filter ופעולת רישום הרכיב בצעו override לפעולה onResume
- בגוף הפעולה הגדירו מסנן ובצעו רישום באופן הבא:

```
@Override
protected void onResume() {

// create the intent filter and
// register receiver dynamically
IntentFilter headSetFilter = new IntentFilter(Intent.ACTION_HEADSET_PLUG);
registerReceiver(receiver,headSetFilter);
```

- חובה לבטל את הרישום! נבצע זאת בפעולה onPause (בצעו override לפעולה)

```
// very important- DO NOT FORGET TO UNREGISTER
@Override
protected void onPause() {
unregisterReceiver(receiver);
```

שלב שלישי - הרצה

- הריצו את האפליקציה עברו למסך שמפעיל את ה broadcast וחברו אזניות למכשיר
- במידה ומריצים על אמולטור - ניתן לבצע חיבור וירטואלי על ידי לחיצה על אפשרויות נוספות- מיקרופון.
- ודאו שחיבור / ניתוק אזניות מציג הודעת Toast על המסך

שלב רביעי- ניתוח שדות ב Intent

- ודאו שהפעולה הגיעה מחיבור או ניתוק של אזניות:

```
if (intent.getAction().equals(Intent.ACTION_HEADSET_PLUG)) {
```

- במידה וכן - יש בהודעה שדות נוספים
- שדות אלו מתוארים ב :android developer
- https://developer.android.com/reference/android/media/AudioManager#ACTION_HEADSET_PLUG
- שדה state שמתאר האם הפעולה היא חיבור או ניתוק
- שדה של microphone שמציין האם לאזניות מיקרופון מובנה או לא
- עדכנו את ה broadcast שלכם שיציג האם הפעולה היא חיבור/ניתוק והאם קיים מיקרופון מובנה באזניות.

```
int state = intent.getIntExtra("state", -1);
int microphone = intent.getIntExtra("microphone",0);
```

יצירת custom broadcast

- לעיתים נרצה לקבל מידע ב activity שלנו מידע שהסתיימה פעולה כלשהי -לדוגמה הורדת קובץ.
- נגדיר Broadcast Receiver עם מסנן שמורכב ממחרוזת שאנחנו נגדיר באפליקציה שלנו
- מכיוון שהודעות יכולות להגיע מאפליקציות אחרות במערכת - יש חשיבות לכך ששם הAction יהיה ייחודי לכן מומלץ לשלב בו את שם האפליקציה
- לדוגמה :

```
"com.example.allon.custom.broadcast"
```

- שאר הפעולות לרישום הרכיב זהות.
- ניתן כמובן לשלוח מידע נוסף ב extra של ה intent
- שימו לב בפעולת onReceive לבדוק שאכן מדובר במחרוזת שהגדרנו

מימוש custom broadcast receiver

- ב Broadcast Receiver ודאו שמדובר בארוע שהגדרנו

```
public class MyCustomReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if(intent.getAction() == "com.example.allon.custom.broadcast")
        {
            // read extra field
            String customData = intent.getStringExtra( name: "data");
```

- ב Activity הגדירו את המסנן - filter בעת יצירת ורישום

```
private MyCustomReceiver receiver;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_send_custom_broadcast);
     receiver = new MyCustomReceiver();
}
@Override
protected void onResume() {
    super.onResume();
    IntentFilter filter = new IntentFilter( action: "com.example.allon.custom.broadcast");
    registerReceiver(receiver,filter);
}
@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(receiver);
}
```

שליחת הודעת Broadcast

- שליחת הודעת broadcast מתבצעת באמצעות intent
- בדוגמה שלנו מדובר בשליחת הודעה מקומית- כלומר בתוך האפליקציה שלנו
- הגדירו intent עם Action בהתאם למחרוזת שיצרנו

```
Intent intent = new Intent("com.example.allon.custom.broadcast");
```

- ניתן לצרף מידע ל intent בשדה ה:extra

```
intent.putExtra("data","my broadcast");
```

- שליחת ההודעה:

```
sendBroadcast(intent);
```

- שימו לב: בניגוד ל StartActivity שליחת broadcast לא מעבירה אותנו ל activity אחר.
- Broadcast Receiver מקבל הודעה, מבצע מספר פעולות ומסיים את תפקידו

עבודה נכונה Broadcast Receiver

- כשמתקבל Broadcast חשוב לבצע פעולה קצרה ב onReceive ולסיים, לאחר שהפעולה חוזרת התהליך מסתיים. לכן הימנעו מהפעלת thread או service בעת קבלת ה broadcast.
- ניתן להרשם עם broadcast ל AlarmManager לקבלת הודעה בזמן מסויים
  - קיים באפליקציית הדוגמה
- השתמשו ב broadcast סטטי (ה filter ACTION מוגדר במניפסט) רק כשנדרש - למשל כשיש חשיבות לפעולה שתתבצע כאשר מכשיר הטלפון עולה לאחר פעולת Restart:

```
android.intent.action.BOOT_COMPLETED
```

- לא להתחיל activity מתוך Broadcast Receiver, אם יש צורך השתמשו במנגנון ה notification
  - קיים באפליקציית הדוגמה

Context

	Application	Activity	Service	ContentProvider	BroadcastReceiver
Show a Dialog	NO	YES	NO	NO	NO
Start an Activity	NO[1]	YES	NO[1]	NO[1]	NO[1]
Layout Inflation	NO[2]	YES	NO[2]	NO[2]	NO[2]
Start a Service	YES	YES	YES	YES	YES
Bind to a Service	YES	YES	YES	YES	NO
Send a Broadcast	YES	YES	YES	YES	YES
Register BroadcastReceiver	YES	YES	YES	YES	NO[3]
Load Resource Values	YES	YES	YES	YES	YES

https://possiblemobile.com/2013/06/context/

הרשאות: Broadcast Receiver

- נדרשות הרשאות בהתאם למדיניות ההרשאות של Android
- ניתן לצרף הרשאה בעת שליחת הודעת Broadcast
- רק אפליקציות שמבקשות את ההרשאה יקבלו את ההודעות
- לטובת קבלת הודעת SMS נדרשת ההרשאה הבאה:

```
>uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

- בעת שליחת ההודעה מצרפים את ההרשאה כך שרק אפליקציות שקיבלו את ההרשאה ונרשמו לארוע יקבלו את הודעת ה Broadcast