

פרק 19 - Service

מה זה Service?

Service הוא שירות שנועד לביצוע פעולות ארוכות טווח.

מה ההגדרה של ארוכות טווח?

טווח של זמן בו יש סיכוי שהמשתמש יעבור לאפליקציה אחרת.

למה לא להשתמש ב Thread?

כשנעבור מסך ה Thread יופסק. כלומר כשנסגור את ה activity ה thread יופסק.

לעומת זאת, ה Service ימשיך לפעול גם כשנעבור בין מסכים.

יש שתי קטגוריות ל-service:

startService

1. יש לו תוחלת חיים משל עצמו.
 2. מוגבל ביכולת שלו לבוא באינטראקציה עם components (מבחינת UI)
 3. רוב ה-services הם started
- שימוש ל StartService: שמיעת מוזיקה ברקע בזמן שמשחקים משחק באפ אחר.

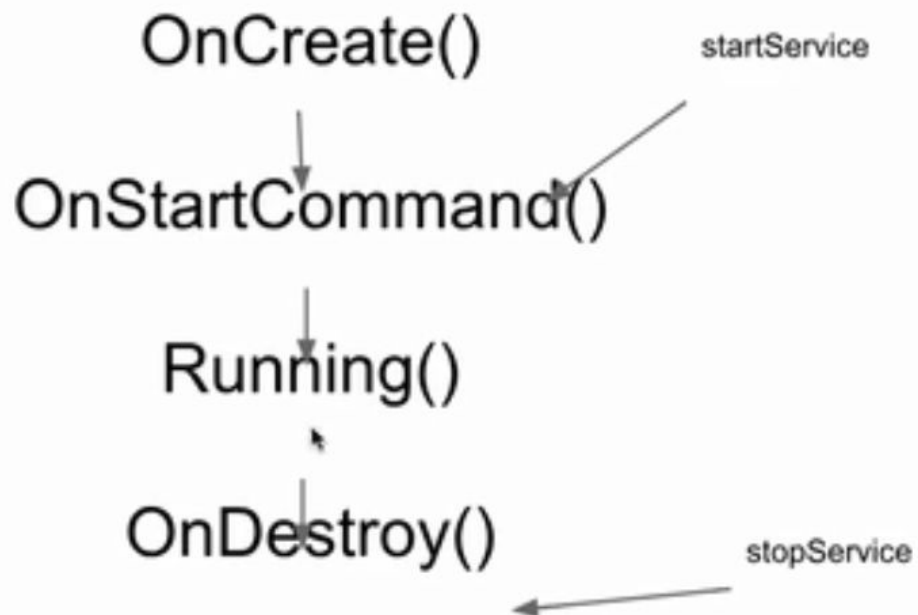
bound

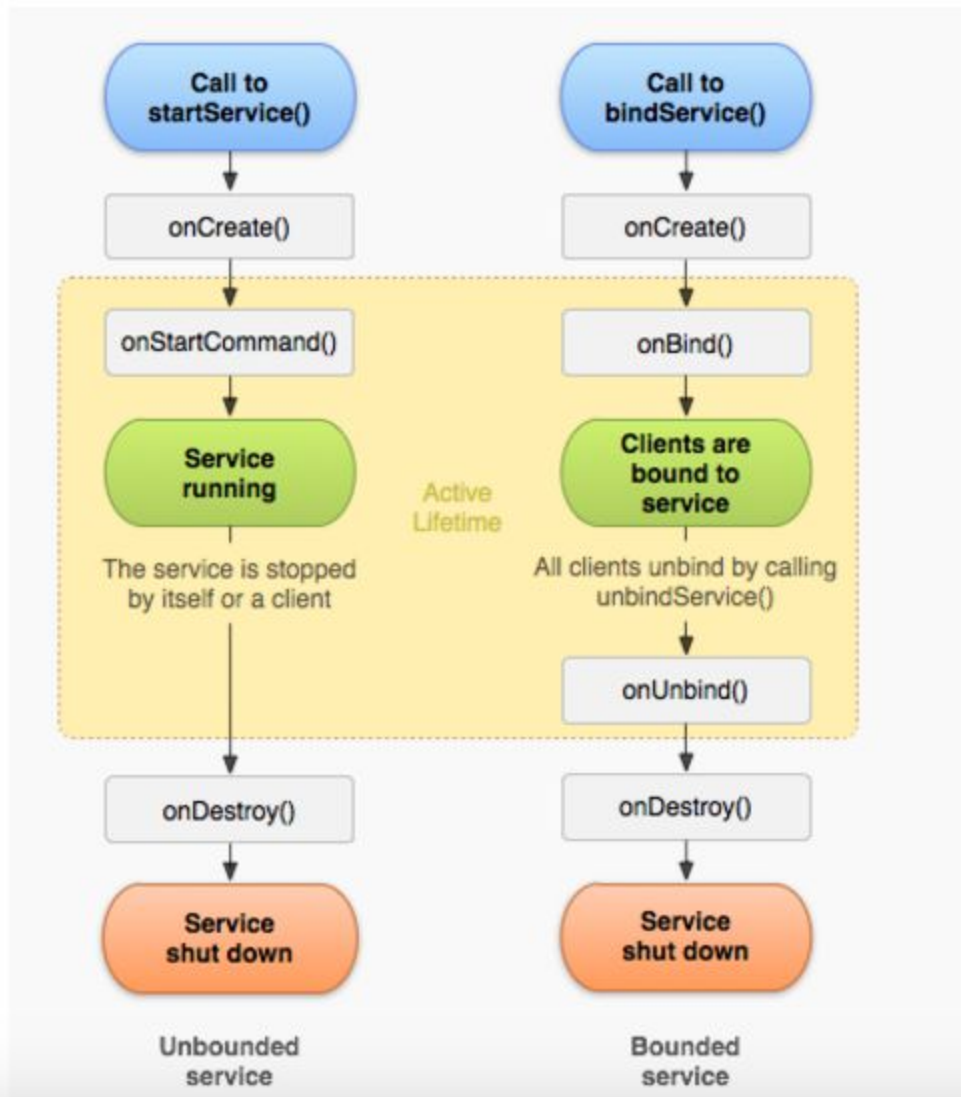
1. תוחלת החיים תלויה ברכיב שיצר אותו
2. יש לו אינטראקציה עם הרכיב שיצר אותו
3. חי כל עוד הרכיב שיצר אותו חי

שרות יכול להיות Started או Bound, זה תלוי אם הפעלנו אותו על ידי onStartCommand (מתאים ל - started) או על ידי onBind מתאים ל Bound

בקורס שלנו נעבור על ה `StartService`.

מעגל החיים של service started





דוגמה מספר 1 ל Service:

שלב 1 - ניצור class FirstService שיורש מ - Service:

```
public class FirstService extends Service {
    FirstThread firstThread;

    String serviceName=FirstService.class.getSimpleName();

    @Override
    public void onCreate() {
        super.onCreate();
        Toast.makeText(this,"service onCreate: " + serviceName,Toast.LENGTH_LONG).show();
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        Toast.makeText(this,"service onStartCommand: " + serviceName,Toast.LENGTH_LONG).show();
        if(firstThread==null){

            firstThread=new FirstThread();
            firstThread.start();

        }

        return super.onStartCommand(intent, flags, startId);
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        firstThread=null;
        Toast.makeText(this,"service onDestroy: " + serviceName,Toast.LENGTH_LONG).show();
    }

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

הצהרה על המשתנים

נוכל לקבל את השם שהקצתה התוכנית ל Service

כשניצור את ה Service נדפיס את שמו ב Toast

כשנפעיל את ה Service יודפס לנו Toast שאומר שהופעל ה Service

ניצור Thread חדש ונפעיל אותו

כשנהרוס את ה Service נגדיר את ה Thread להיות null. בנוסף נדפיס בעזרת Toast שהרסנו את ה Service

```
public class FirstThread extends Thread{
    int counter=30;
```

```
@Override
public void run() {
    super.run();
```

```
while(counter>0){
    try {
        Thread.sleep(1000);
        Log.d("service", "timer : " + counter);
        counter--;
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
stopSelf();
}
```

נממש את הקלאס שירש מ Thread
 בתוך הקובץ של FirstThread
 :FirstService

השעון שלנו ירוץ אחורה
 ויתחיל מ 30. כל שנייה
 יודפס ערך ה counter
 ונוריד אותו ב 1.

כשנגיע ל 0 יעצר ה Thread

שלב 2 - עיצוב ה MainActivity:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```

```
<TextView
```

```
    android:text="@string/hello_world"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<Button
```

```
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/btnStart"
    android:text="start"
```

```

/>
<Button
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/btnStop"
    android:text="stop"

/>
</LinearLayout>

```

שלב 3 - MainActivity:

```

public class MainActivity extends AppCompatActivity{
    Button btnStart,btnStop;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnStart=(Button)findViewById(R.id.btnStart);
        btnStart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this,FirstService.class);
                startService(intent);
            }
        });
        btnStop=(Button)findViewById(R.id.btnStop);
        btnStop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this,FirstService.class);
                stopService(intent);
            }
        });
    }
}

```

נצהיר על כפתורים

בלחיצה על הכפתור btnStart

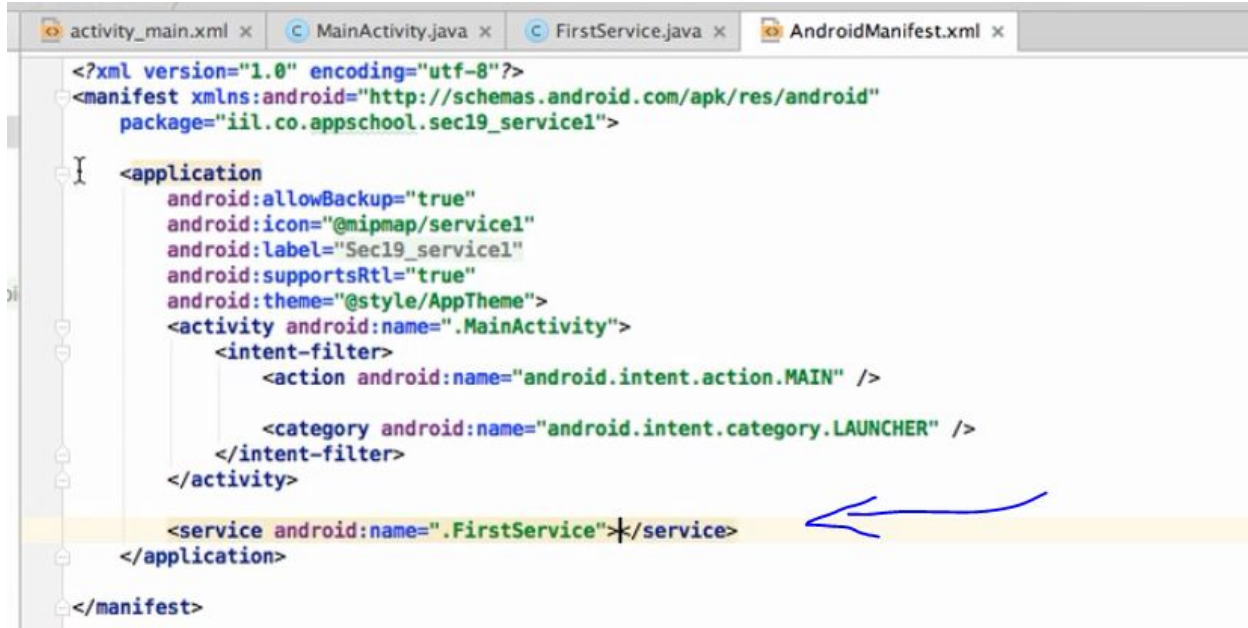
ניצור intent שיפעיל את ה service

נפעיל את ה intent

בלחיצה על btnStop

נעצור את ה Service

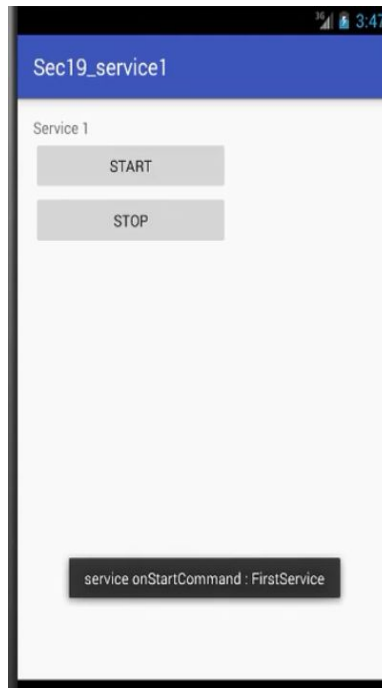
שלב 4 - שמירה במניפסט:



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="iil.co.appschool.sec19_service1">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/service1"
        android:label="Sec19_service1"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service android:name=".FirstService"*/>
    </service>
    </application>
</manifest>
```

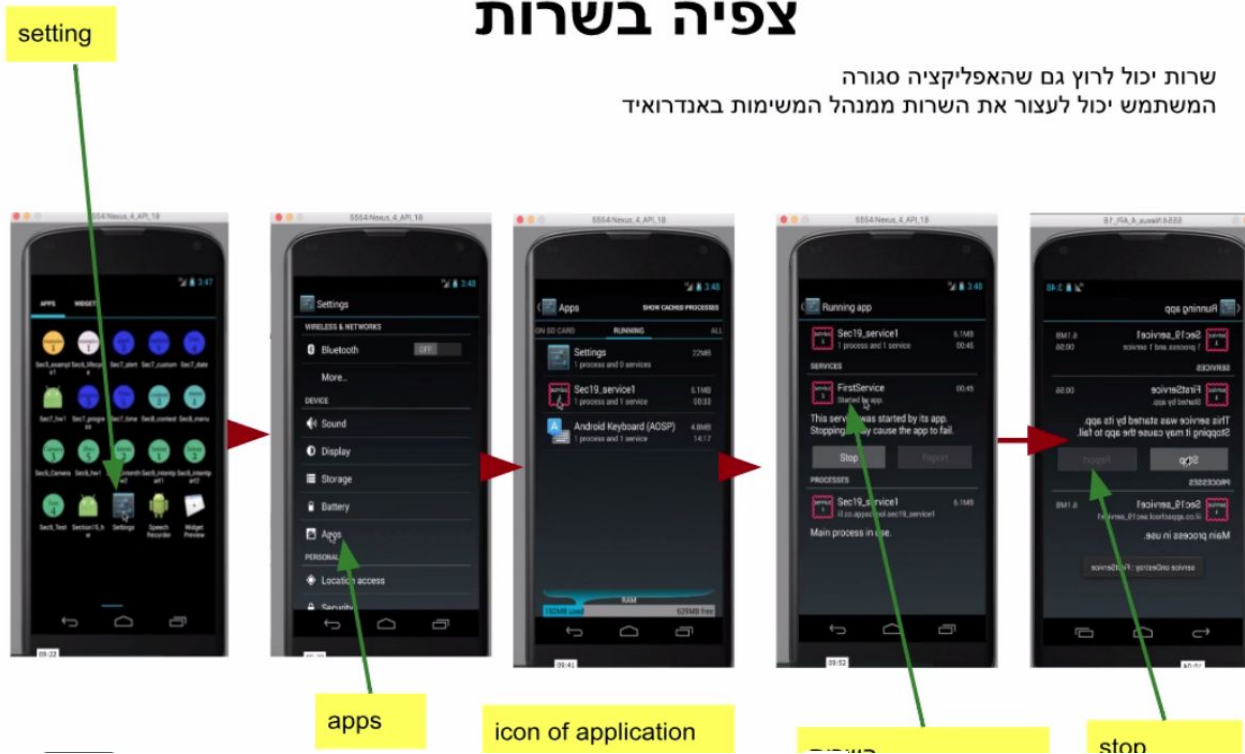
תוכלו להריץ כעת את האפליקציה, ולראות את ה Service שלכם
בזמן ריצה (:



במידה ולא הגדרתם כפתור לעצירת ה service באפליקציה, תוכלו לעצור דרך ה
:Settings

צפיה בשרות

שרות יכול לרוץ גם שהאפליקציה סגורה
 המשתמש יכול לעצור את השרות ממנהל המשימות באנדרואיד



Intent Service

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 `StopSelf`

לעומת זאת `startService` - רץ על ה-`Thread` הראשי.

יצירת `IntentService`:

1. יצירת class שיורש מ- `IntentService`

2. ב- `constructor` יש לתת שם ל- `thread`

3. דריסה של הפונקציה `IntentService.onHandleIntent`

4. רישום ה-`service` ב- `manifest`

דוגמה לשעון שרץ אחורנית בעזרת IntentService:

שלב 1 - עיצוב ה MainActivity:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:text="Intent Service"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:id="@+id/btnStart"
        android:text="start"
        />

    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:id="@+id/btnStop"
        android:text="stop"
        />
</LinearLayout>
```

שלב 2 - יצירת ה IntentService:

ניצור מחלקה שיורשת מ - IntentService וממשת את onHandleIntent:

```
public class SecondService extends IntentService {

    public SecondService() {
        super("secondservice");
        Log.d("service", "constructor second service called");
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        Log.d("service", "onHandleIntent secondservice called");
        int counter = intent.getExtras().getInt("counter");
        while(counter > 0) {
            try {
                Thread.sleep(1000);
                counter--;
                Log.d("service", "counter=" + counter);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        Toast.makeText(this, "destory service", Toast.LENGTH_LONG).show();
    }
}
```

ירושה מ IntentService

בבנאי נעביר את שם ה thread

בלולאת while ירד ערך ה counter ב 1 כל שנייה עד שנגיע ל 0. נדפיס את ערך ה counter בקונסולה

חובה לשלוח שם ל Thread

מופע חדש של MyHandler

שלב 3 - MainActivity:

```
public class MainActivity extends ActionBarActivity {
    Button btnStart, btnStop;
    Intent intent;
```

← הצהרות

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
    btnStart=(Button)findViewById(R.id.btnStart);
```

```
    btnStart.setOnClickListener(new View.OnClickListener() {
```

@Override

```
    public void onClick(View v) {
```

```
        intent=new Intent(MainActivity.this,SecondService.class);
```

```
        intent.putExtra("counter",30);
```

```
        startService(intent);
```

```
    }
```

```
});
```

```
    btnStop=(Button)findViewById(R.id.btnStop);
```

```
    btnStop.setOnClickListener(new View.OnClickListener() {
```

@Override

```
    public void onClick(View v) {
```

```
        intent=new Intent(MainActivity.this,SecondService.class);
```

```
        stopService(intent);
```

```
    }
```

```
});
```

```
}
```

בלחיצה על btnStart ניתן הפניה ל
Intent. נעביר ב Extras של ה intent
את מספר השניות שנרצה שירץ ה
counter. נפעיל את ה Intent

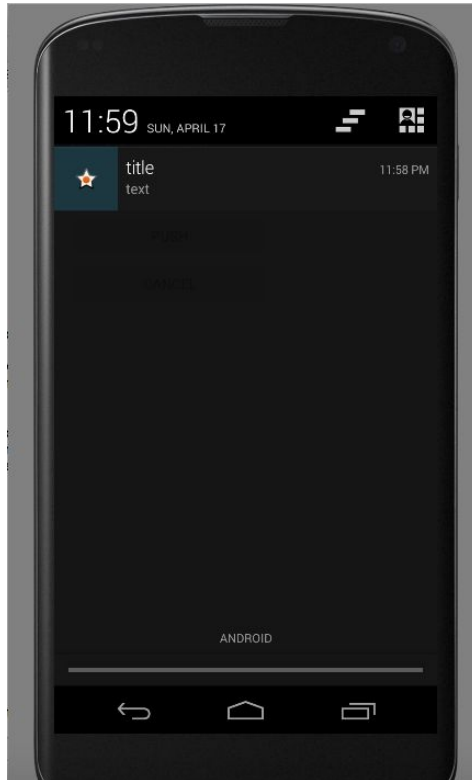
בלחיצה על btnStopניתן הפניה
ל Intent ונעצור את ה Intent

זהו זה! הריצו את האפליקציה (:

Status Bar Notification

מה זה Status Bar Notification?

המסך שמופיע כשאתם גוררים את השורה העליונה שבמסך שלכם כלפי מטה:



ה Notification Manager הוא אובייקט שמטרתו לנהל את ההודעות שמופיעות ב Status Bar Notification.

ה Notification Manager חייב לקבל שלושה פרמטרים:

- Icon
- Title
- pendingIntent

יש אפשרות להוסיף פרמטרים נוספים:

- Ticket text

- sound
- vibrate
- flashing led

PendingIntent:

בדומה ל intent שדרכו אנחנו יכולים לעבור בין שני Activity ולהפעיל Service. ה Pending Intent הוא:

מסר שניתן ליישום אחר לביצוע קטע קוד שמוגדר מראש.

ניתן ליצור PendingIntent במספר דרכים

```
getActivity(Context c,int,Intent,int)  
getBroadcast(Context,int,Intent,int)  
getService(Context,int,Intent,int)
```

נוכל להעלים הודעה בעזרת אחת משתי הדרכים:

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

```
public void cancelAll()
```

נוכל גם להוסיף רטט להודעה:

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

העברת מיקום קובץ

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

vibrate: - צריך להוסיף במניפסט הרשאה ל

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

נוכל להוסיף קול להודעה:

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

נוכל להוסיף מוזיקה ולהגדיר אותה כצליל ההודעה: (במקום ה default)

העברת מיקום קובץ

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

העברת מיקום משאב בפרויקט

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

הקול יופעל עד שההודעה תטופל :

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

דגלים שימושיים שניתן להעביר:

FLAG_NO_CLEAR - הודעה שלא מתבטלת

FLAG_ONGOING_EVENT - הודעה במהלך אירוע

FLAG_AUTO_CANCEL - הודעה שתבטל שיגעו בה -

FLAG_INSISTENT - קול שיופיע עד שההודעה תטופל -

דוגמה לשימוש ב Status Bar Notification:

שלב 1 - עיצוב ה MainActivity:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
tools:context=".MainActivity">
```

```
<TextView  
    android:text="Notification manager"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

```
<Button  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:id="@+id/btnPush"  
    android:text="push"  
>
```

```
</Button>
```

```
<Button  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:id="@+id/btnCancel"  
    android:text="cancel"  
/>
```

```
</LinearLayout>
```

שלב 2 - עיצוב ה SecondActivity שאליו נגיע אם נלחץ על ההודעה:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context="com.apps.appschool.notificationmanagerpart1.SecondActivity">
```

Copyright © 2019 appSchool. Powered by appSchool.co.il

```

<TextView
    android:text="Notification manager"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tv"
    android:textSize="30sp"
/>
  
```

```
</RelativeLayout>
```

שלב 3 - SecondActivity:

```
public class SecondActivity extends AppCompatActivity {
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_second);
```

```
    TextView tv= (TextView)findViewById(R.id.tv);
```

```
    Intent intent=getIntent();
```

```
    String str= intent.getExtras().getString("key");
```

```
    tv.setText(str);
```

```
}
```

```
}
```

ניתן הפניה ל TextView

נקבל הפניה ל intent
שהועבר למסך הזה

נשלוף את הערך שהעברנו
ב Extras עם ה "key"

נשנה את הטקסט שב
TextView להיות הטקסט שקיבלנו מה
Extras

שלב 4 - MainActivity:

```
public class MainActivity extends AppCompatActivity {
    Button btnCancel, btnPush;
```

הצהרות

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
    btnCancel = (Button) findViewById(R.id.btnCancel);
    btnPush = (Button) findViewById(R.id.btnPush);
    btnPush.setOnClickListener(new View.OnClickListener() {
```

@Override

```
    public void onClick(View v) {
```

//phase 1

```
        int icon = android.R.drawable.star_on;
        String ticket = "this is ticket message";
        long when = System.currentTimeMillis();
        String title = "title";
        String ticker = "ticker";
        String text = "text";
```

//phase 2

```
        Intent intent = new Intent(MainActivity.this, SecondActivity.class);
        intent.putExtra("key", "Uzi oranim");
        PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
        NotificationManager notificationManager = (NotificationManager)
            getSystemService(Context.NOTIFICATION_SERVICE);
```

בלחיצה על btnPush:

נגדיר את התמונה שבהמשך
תקפוץ יחד עם ההודעה (בעזרת
שמירת הכתובת שלה ב int)

נגדיר את הטקסטים שיופיעו בהודעה

נגדיר את ה intent שיעביר
אותנו ל SecondActivity

נעביר ב Extras את השם שנרצה
SecondActivity בטקסט ב

נגדיר שה intent שיופעל בלחיצה
על ההודעה יהיה ה intent שיצרנו

//phase 3

```
        NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext());
        Notification notification = builder.setContentIntent(pendingIntent)
            .setSmallIcon(icon).setTicker(ticker).setWhen(when)
            .setAutoCancel(true).setContentTitle(title)
            .setSmallIcon(android.R.drawable.star_on)
            .setContentText("content text").build();
        notificationManager.notify(1, notification);
```

ניתן ל Notification את ה pendingIntent שיצרנו

נוסיף את התמונה, הטקסטים,
ומספר השניות שיקח להודעה
להופיע שהגדרנו

```
    }
```

```
});
```

```
btnCancel.setOnClickListener(new View.OnClickListener() {
```

@Override

```
    public void onClick(View v) {
```

```
        NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.cancel(1);
```

```
    }
```

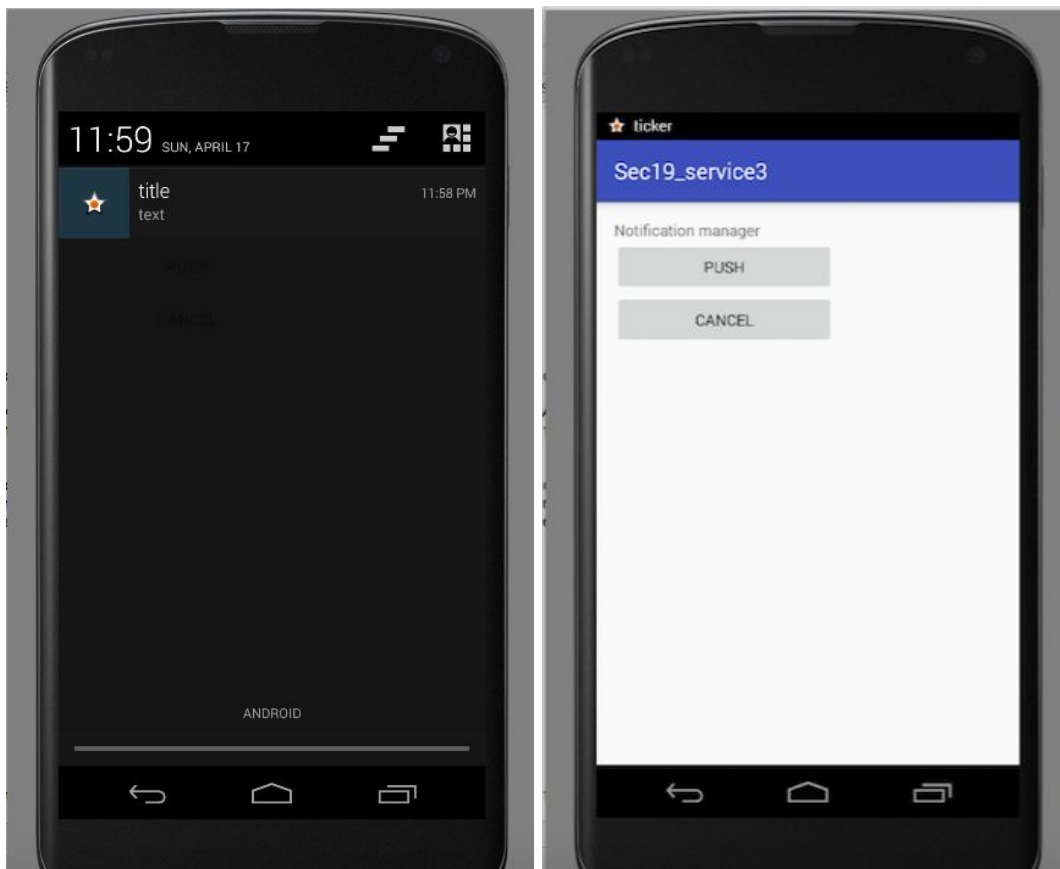
```
});
```

```
}
```

```
}
```

בלחיצה על btnCancel נסגור
את ה NotificationManager

סוף הדוגמה (:



Foreground (שירות חזית)

1. שירות חזית מספק הודעה אשר ממוקמת תחת OnGoing
2. כדי להפעיל את ה-Service צריך לקרוא ל-`startForeground()`
3. הפונקציה `StartForGround` מקבלת שני פרמטרים: מספר מזהה, ואת ההודעה שבשורת המצב.
4. על מנת לבטל `forGround` קוראים לפונקציה `stopForGround(true)`
דוגמה לשימוש ב `Foreground`:

שלב 1 - עיצוב MainActivity:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```

```
<TextView
    android:text="Foreground Service"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<Button
```

```
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/btnPush"
    android:text="push"
    >
```

```
</Button>
```

```
<Button
```

```
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/btnCancel"
    android:text="cancel"
    />
```

```
</LinearLayout>
```

שלב 2 - עיצוב ה SecondActivity שאליו נגיע אם נלחץ על ההודעה:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
tools:context="com.apps.appschool.notificationmanagerpart1.SecondActivity">
```

```
<TextView
```

```
    android:text="Notification manager"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/tv"
```

```
    android:textSize="30sp"
```

```
/>
```

```
</RelativeLayout>
```

שלב 3 - SecondActivity:

```
public class SecondActivity extends AppCompatActivity {
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_second);
```

```
    TextView tv= (TextView)findViewById(R.id.tv);
```

```
    Intent intent=getIntent();
```

```
    String str= intent.getExtras().getString("key");
```

```
    tv.setText(str);
```

```
}
```

```
}
```

ניתן הפניה ל TextView

נקבל הפניה ל intent
שהועבר למסך הזה

נשלוף את הערך שהעברנו
ב Extras עם ה "key"

נשנה את הטקסט שב
TextView להיות הטקסט שקיבלנו מה Extras

שלב 4 - יצירת Service:

```
public class PushService extends Service {
```

```
    @Override
```

```
    public IBinder onBind(Intent intent) {  
        return null;  
    }
```

```
    @Override
```

```
    public int onStartCommand(Intent intent, int flags, int startId) {  
        startForeground(1, getNotification());  
        Toast.makeText(getApplicationContext(), "startForeground", Toast.LENGTH_LONG).show();  
        return super.onStartCommand(intent, flags, startId);  
    }
```

שני הפרמטרים:
מספר מזהה, הודעה
בשורת המצב

```
    @Override
```

```
    public void onDestroy() {  
        super.onDestroy();  
        Toast.makeText(getApplicationContext(), "stopForeground", Toast.LENGTH_LONG).show();  
  
        stopForeground(true);  
    }
```



```
public Notification getNotification(){
```

```
//phase 1
```

```
int icon = android.R.drawable.star_on;  
String ticket = "this is ticket message";  
long when = System.currentTimeMillis();  
String title = "title";  
String ticker = "ticker";  
String text="text";
```

נגדיר את העיצוב של ההודעה

נגדיר את ה intent שיעביר
אותנו ל SecondActivity
בלחיצה על ההודעה

```
//phase 2
```

```
Intent intent = new Intent(getApplicationContext(), SecondActivity.class);  
intent.putExtra("key", "Uzi oranim");  
PendingIntent pendingIntent = PendingIntent.getActivity(getApplicationContext(), 0, intent, 0);
```

נעביר ב Extras את השם שנרצה
שיופיע בטקסט ב SecondActivity

```
//phase 3
```

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext());  
Notification notification = builder.setContentIntent(pendingIntent)  
    .setSmallIcon(icon).setTicker(ticker).setWhen(when)  
    .setAutoCancel(true).setContentTitle(title).  
    setSmallIcon(android.R.drawable.star_on).setDefaults(Notification.DEFAULT_SOUND)  
    .setContentText("content text").build();
```

ניתן להודעה את
העיצוב שהגדרנו

```
//-----you can delete this lines-----
```

```
//NotificationManager notificationManager = (NotificationManager)
```

```
getSystemService(Context.NOTIFICATION_SERVICE);
```

```
//notificationManager.notify(1, notification);
```

```
return notification;
```

```
}  
}
```

שלב 5 - MainActivity:

```
public class MainActivity extends AppCompatActivity {
    Button btnCancel;
    Button btnPush;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
    btnCancel=(Button)findViewById(R.id.btnCancel);
    btnPush=(Button)findViewById(R.id.btnPush);
    btnPush.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, PushService.class);
            startService(intent);
```

```
        }
    });
```

```
    btnCancel.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, PushService.class);
            stopService(intent);
```

```
        }
    });
```

```
}
```

```
}
```

בלחיצה על btnPush ניצור Intent שיפעיל את ה Service ונפעיל אותו

בלחיצה על btnCancel ניצור Intent עם ה Service ונעצור איתו את ה Service

