

Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded

CMSC436: Programming Handheld Systems

Alarms

Today's Topics

Alarms

AlarmManager APIs

Alarm Types

Example Application

Alarms

Mechanism for sending Intents at some point in the future

Allows one application to make code execute, even when that application is no longer running

Alarms

Once registered, Alarms remain active even if the device is asleep

Can configure Alarms to wake a sleeping device

Alarms are canceled on device shutdown/restart

Alarm Examples

MMS - Retry Scheduler

Settings - Bluetooth Discoverable timeout

Phone - User Info Cache

AlarmManager

You create and manage Alarms by interacting with the AlarmManager

Get a reference to the AlarmManager, e.g., by calling the Context class'

```
getSystemService(Context.ALARM_SERVICE)
```

Creating Alarms

```
// One-shot Alarm with inexact timing. If there is already an alarm  
// scheduled for the same IntentSender, that previous alarm will first  
// be canceled
```

```
open fun set(type: Int, triggerAtMillis: Long,  
            operation: PendingIntent!): Unit
```

Creating Alarms

// One-shot Alarm with exact timing

```
open fun setExact(type: Int, triggerAtMillis: Long,  
                 operation: PendingIntent?): Unit
```

On newer API versions, this method generally requires the
SCHEDULE_EXACT_ALARM permission

Creating Alarms

// Repeating alarm with inexact timing

```
open fun setRepeating(type: Int, triggerAtMillis: Long,  
                    intervalMillis: Long, operation: PendingIntent!): Unit
```

Alarm Types

Two degrees of configurability

How to interpret time

What to do if the device is sleeping when the Alarm fires

Interpreting Time

Realtime - relative to system clock

Elapsed - relative to time since last boot up

Sleeping Devices

When Alarm fires and device is asleep, can either

- Wake up device now & deliver Intent

- Wait to deliver Intent until device wakes up

Alarm Type Constants

RTC_WAKEUP

RTC

ELAPSED_REALTIME

ELAPSED_REALTIME_WAKEUP

PendingIntent

A description of an Intent and a target action to perform with it

Can be handed to other applications so that they can perform actions on your behalf at a later time

Key concern - proxy applications shouldn't perform operations that originating application can't

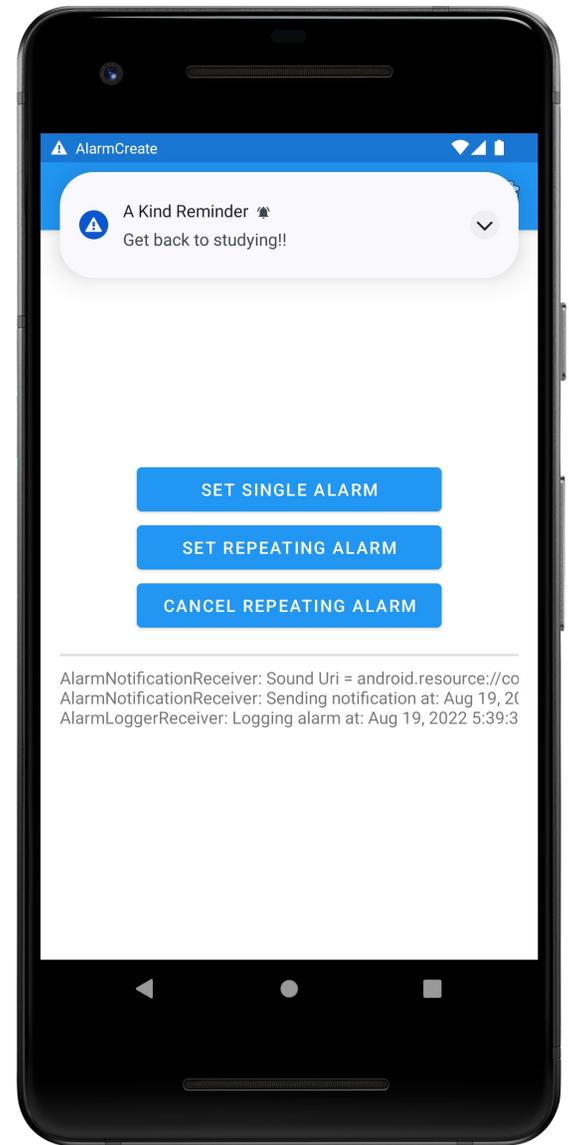
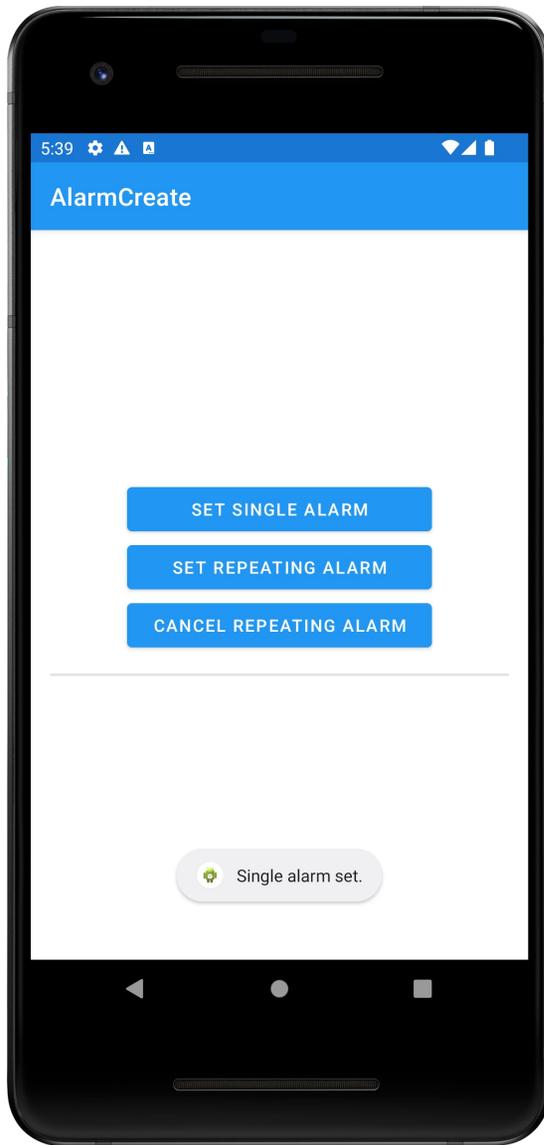
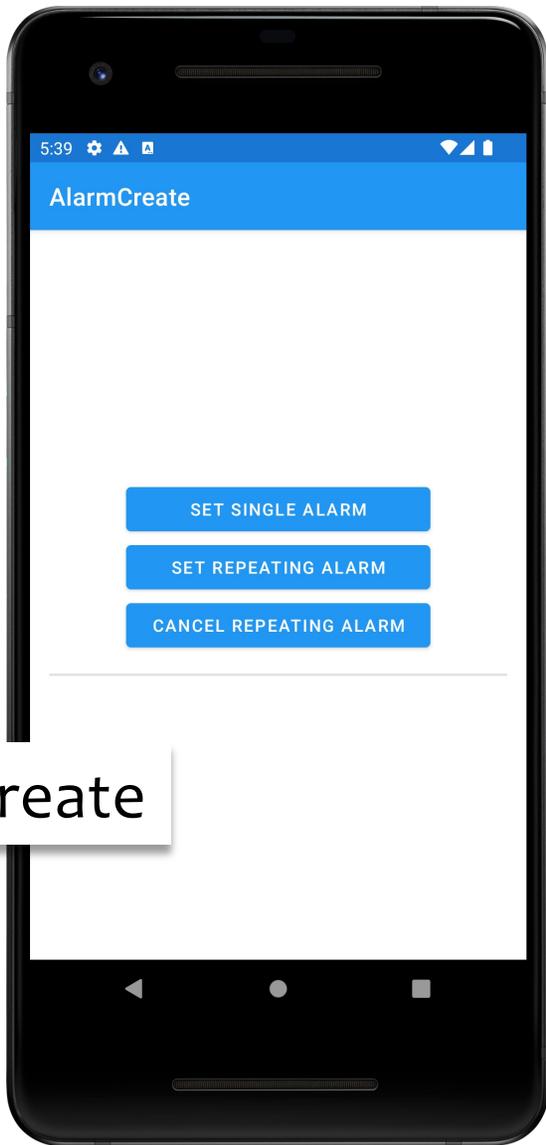
PendingIntent

```
static fun getActivity(context: Context!, requestCode: Int,  
                      intent: Intent!, flags: Int): PendingIntent!
```

```
static fun getBroadcast(context: Context!, requestCode: Int,  
                       intent: Intent!, flags: Int): PendingIntent!
```

```
static fun getService(context: Context!, requestCode: Int,  
                     intent: Intent, flags: Int): PendingIntent!
```

AlarmCreate



Next Time

Threads, Messages and Handlers

Example Applications

AlarmCreate