UNIVERSITÀ
DEGLI STUDI
DE L'AQUILA

disim

Mobile Programming Laboratory

ANDROID
Notifications

Teachers

Ing. Tarquini Francesco, Ph.D

Ph.D in Computer Science Engineering

francesco.tarquini@univaq.it

Ing. D'Errico Leonardo

Ph.D Student in Computer Science Engineering

leonardo.derrico@graduate.univaq.it

Teaching Materials

Available on MOODLE platform

http://www.didattica.univaq.it

Google Drive Repository

https://drive.google.com/drive/folders/1ISqZfn0i9Ub3eWNXbvW00rd0hD9ya8OL?usp=sharing

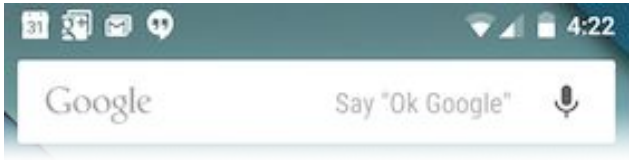
Topics

- Notifications
  - Channel
  - Creating a notification
    - Simple
    - Expanded
  - Managing the notifications
    - Update and Remove
  - Badges
  - Actions

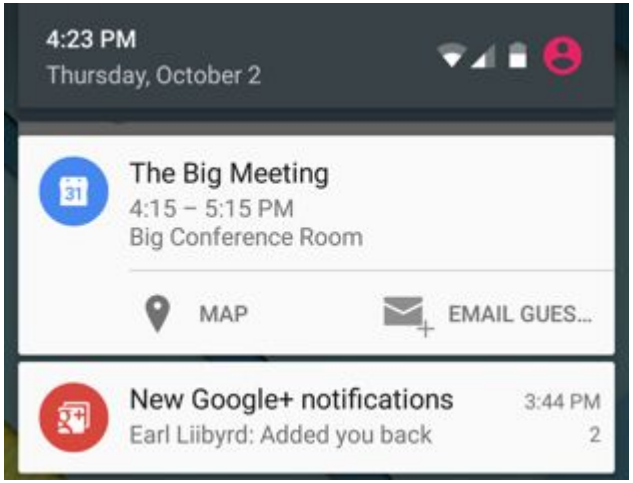
Notifications

A notification is a message you display to the user outside of your app's normal UI.

The notification area and notification drawer are system-controlled areas



Notifications area



Notifications drawer

Notifications - Channel

Starting Android 8.0 (API 26), notification channels allows you to create a user-customizable channel.

When you target API 28, you **must** implement one or more notification channels to display notifications to your users.

You can create an instance of **NotificationChannel** for each distinct type of notification you need to send.

Using the channel, the users can manage most of the settings associated with notification

- Importance
- Sound
- Lights
- Vibration
- Show on lockscreen
- Override do not disturb

Notifications - Channel

To create a notification channel:

Create an ID that’s unique within your package

```
String id = "notification_channel_id";
String name = "Notification Name";
int importance = NotificationManager.IMPORTANCE_DEFAULT;
NotificationChannel channel = new NotificationChannel(id, name, importance);
```

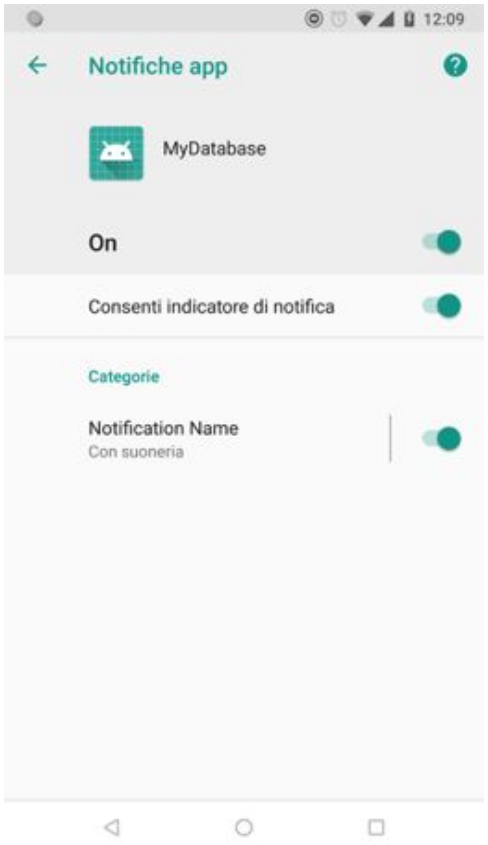
Configure the channel object

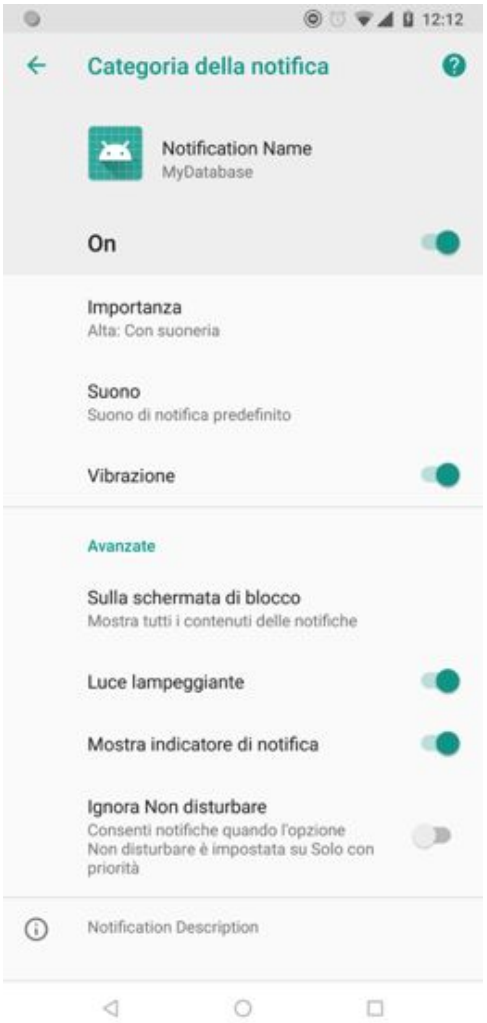
```
channel.setDescription("Notification Description");

// Lights
channel.enableLights(true);
channel.setLightColor(Color.RED);

// Vibration
channel.enableVibration(true);
channel.setVibrationPattern(new long[]{ 100, 200, 300, 400, 500, 600 });

// Sound
AudioAttributes audioAttrs = new AudioAttributes.Builder()
      .setUsage(AudioAttributes.USAGE_ALARM).build();
Uri uri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
channel.setSound(uri, audioAttrs);
```

Submit the channel to the notification manager

```
NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
if(manager != null) manager.createNotificationChannel(channel);
```

Notifications - Channel





Notifications - Creating a notification

A Notification object must contain the following:

- A small icon
- A title
- Detail text
- On Android 8.0 and higher, a valid notification channel ID

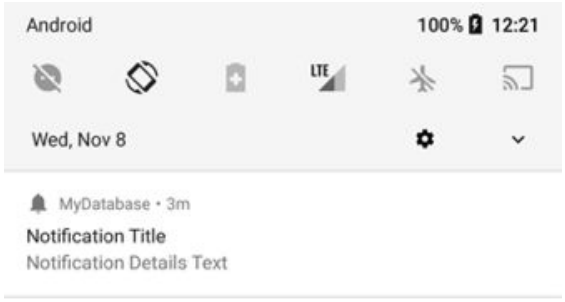
A notification can provide multiple actions:

- usually the action on the user click opens an Activity in your app
- you can add specific buttons to the notification

Notifications - Creating a notification

The following code illustrates a simple notification that specifies an activity to open when the user clicks the notification.

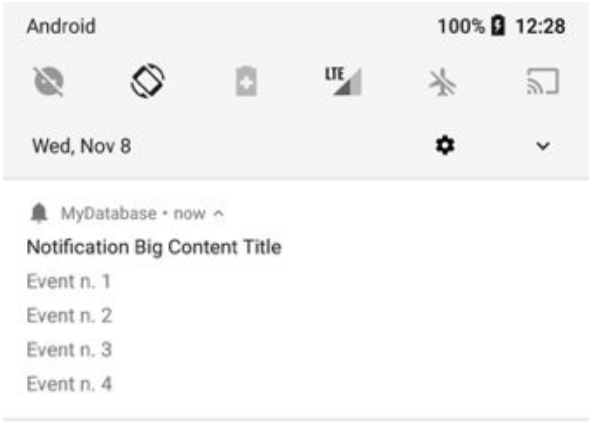
```
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(), channelId)
        .setSmallIcon(R.drawable.ic_notifications)
        .setContentTitle("Notification Title")
        .setContentText("Notification Details Text");

Intent intent = new Intent(this, MainActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pending = PendingIntent.getActivity(this, 0, intent, 0);

builder.setContentIntent(pending);
NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
if (manager != null) manager.notify(NOTIFICATION_ID, builder.build());
```



Notifications - Creating a notification

The following code demonstrates how to alter the notification created to use the expanded layout

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(), channelId)
      .setSmallIcon(R.drawable.ic_notifications)
      .setContentTitle("Notification Title")
      .setContentText("Notification Details Text");

NotificationCompat.InboxStyle inboxStyle = new NotificationCompat.InboxStyle();
inboxStyle.setBigContentTitle("Notification Big Content Title");
for(int i = 0; i < 4; ++i){
   inboxStyle.addLine("Event n. " + (i+1));
}
builder.setStyle(inboxStyle);
```



Notifications - Managing the notifications

Android gives us API to update a notification and remove it

You always use notify() method to update notification using the same id

```
NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
if (manager != null) manager.notify(NOTIFICATION_ID, builder.build());
```

Notifications remain visible until one of the following happens:

- The user dismisses the notification
- The user clicks the notification, and you called **setAutoCancel**()
- You call **cancel**() for a specific notification ID

```
NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
if (manager != null) manager.cancel(NOTIFICATION_ID);
```

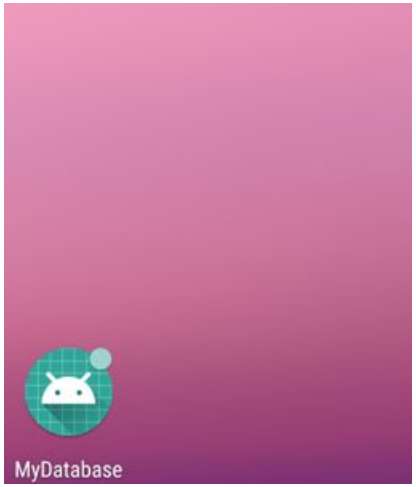
- You call **cancelAll**()
- If you set a timeout using **setTimeoutAfter**()

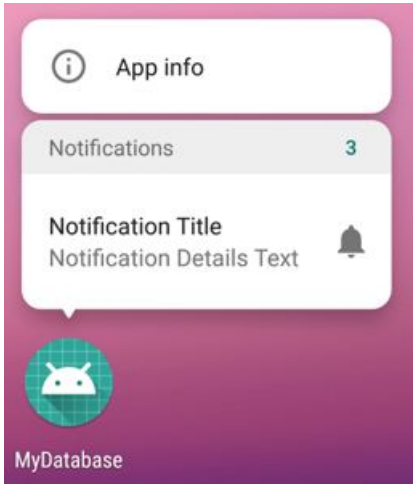
Badges

Since Android 8.0 you can add a badge at the app's icon to show that the app have notifications.

It is very easy to add a badge, you can add an option to created channel using **setShowBadge()** method.

```
// Badges
channel.setShowBadge(true);
```





Actions

You can apply an action on the notification

```
builder.addAction(R.drawable.ic_notifications, "Open", pending);
```