

Notifications

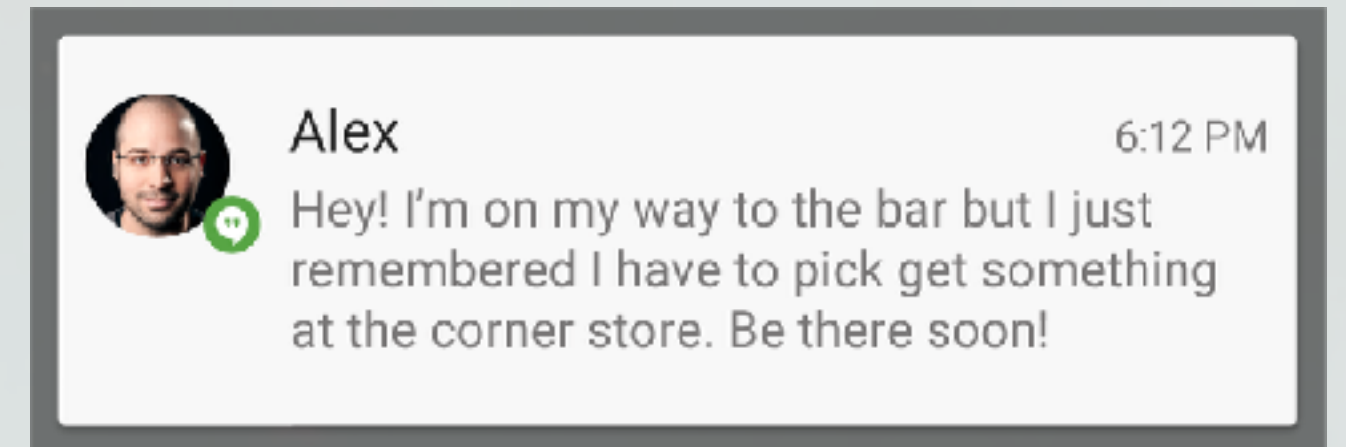
Renault@lrde.epita.fr





Inform the User that something

- 🔊 just happened (SMS, mail, etc)
- 🔊 will happen (reminders, meetings, etc.)



Only visible for a certain amount of time

- 🔊 they do not stop the current activity
- 🔊 they use dedicated (restricted) space for their display
- 🔊 the user is free to interrupt its activity through an action



Lollipop (and higher version) provide new notifications that are more aesthetic and allow more actions

Relative Importance of Notifications

MAX

Critical: something requires an immediate attention

HIGH

Communication: SMS, mail, telephony, ...

DEFAULT

All others: games,

LOW

Non-important: Software upgrades

MIN

Contextual informations: weather, etc.

Relative Importance of Notifications

MAX

Critical: something requires an immediate attention

HIGH

Communication: SMS, mail, telephony, ...

Notifications are ordered according to their priorities

All others: games,

LOW

Non-important: Software upgrades

MIN

Contextual informations: weather, etc.

Relative Importance of Notifications

MAX

Critical: something requires an immediate attention

HIGH

Communication: SMS, mail, telephony, ...

Notifications are ordered according to their priorities

Your application is not more important than all the other ones

LOW

Non-important: Software upgrades

MIN

Contextual informations: weather, etc.

Manage Notifications (1/2)

Building a notification

```
NotificationCompat.Builder mBuilder =  
    new NotificationCompat.Builder(getBaseContext())  
        .setSmallIcon(R.drawable.ic_launcher)  
        .setContentTitle("My notification")  
        .setContentText("Hello World!")  
        .setPriority(Notification.PRIORITY_MAX);
```

Grab the notification manager

```
// Gets an instance of the NotificationManager service  
NotificationManager mNotifyMgr = (NotificationManager)  
    getSystemService(NOTIFICATION_SERVICE);
```

Publish it:

```
mNotifyMgr.notify(mNotificationId, mBuilder.build());
```

For later updates

Manage Notifications (2/2)

Suppress a notification

-  cancel(int): delete a notification using its identifier
-  cancelAll(): delete all notifications from the current application

Fix the visibility on the home screen

-  setVisibility: fix the visibility using above criteria

VISIBILITY_PUBLIC	Display all the content of the notification
VISIBILITY_PRIVATE	Only display basic informations (application, etc.)
VISIBILITY_SECRET	Minimum display: not even the name of the originated application

Toasts



Toast

Contextual
Message



- Immediate message popping like a toast



```
Toast.makeText(getApplicationContext(),  
    "MyText", Toast.LENGTH_SHORT).show();
```

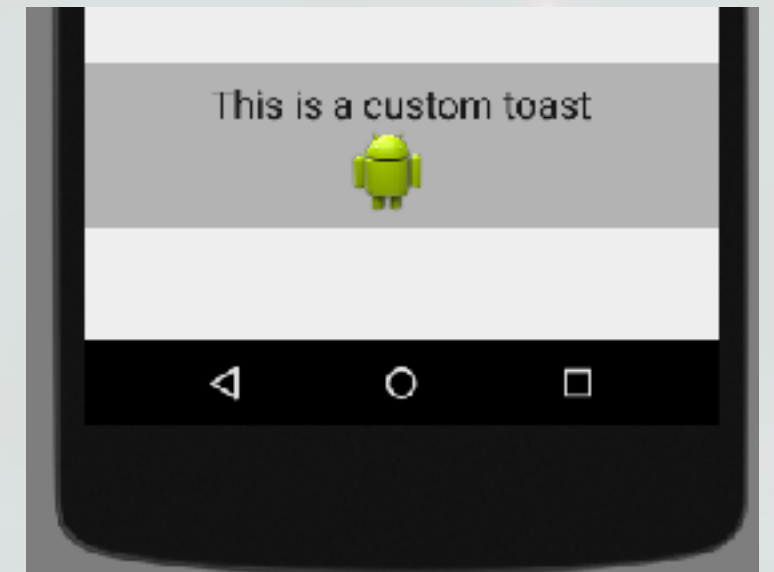
- The length of the display can be fixed
 - ▶ **Toast.LENGTH_SHORT: 2 secondes**
 - ▶ **Toast.LENGTH_LONG: 3.5 secondes**
- The position over the screen can be fixed using `setGravity`
 - ▶ **Gravity.TOP, Gravity.BOTTOM, Gravity.RIGHT, Gravity.LEFT**

```
toast.setGravity(Gravity.TOP | Gravity.LEFT, xoffset, yoffset);
```


Customized Toasts

Toasts can be customized

-  Define a layout (here TextView + ImageView)
-  Use the layout inflater to build it







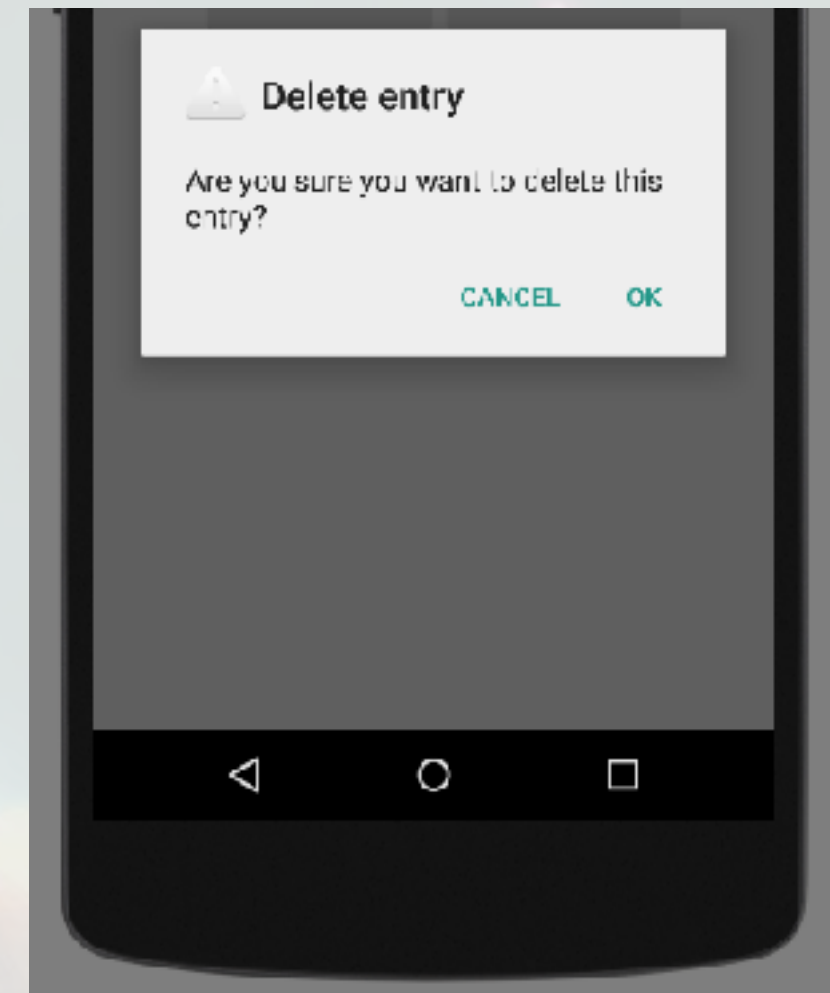
```
LayoutInflater inflater = getLayoutInflater();
View layout = inflater.inflate(R.layout.toast_layout,
    (ViewGroup) findViewById(R.id.toast_layout_root));
TextView text = (TextView)
    layout.findViewById(R.id.textDisplay);
text.setText("This is a custom toast");
Toast toast = new Toast(getApplicationContext());
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);
toast.show();
```

-  Use constructor only if there is a View in the customized Toast!
Otherwise makeText if enough!

AlertDialog

Generic user interaction

-  A title
-  Three button maximum
 - ▶ **with customizable actions**
-  Clickable list of elements
-  Customized Layout



```
new AlertDialog.Builder(MainActivity.this)
    .setTitle("Delete entry")
    .setMessage("Are you sure you want to delete this entry?")
    .setIcon(android.R.drawable.ic_dialog_alert)
    .setPositiveButton(android.R.string.yes,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                // continue with delete
            }
        }
    ).show();
```

Snackbar

Non-intrusive notifications

-  One button (i.e. action) maximum
-  Customizable



Build.gradle may be updated

```
compile 'com.android.support:design:23.1.1'
```

```
Snackbar.make(findViewById(R.id.container),  
    "Item 9 pressed", Snackbar.LENGTH_INDEFINITE)  
    .setAction("close", new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            }  
    }).show();
```

Summary



Many components can be used to interact with the user without building a with with dedicated buttons

it helps to have a homogeneous environment across applications



Notifications warn that something happened

Priority can be fixed



Contextual notifications

Toast: transient information

Dialog: require a user action



Other kind of notification exist

For instance, ProgressView notify that something is actually in progress



