

# Notification avec commandes

Par Souad BELLOUNI



**Android**  
Studio

# Travail demandé

- Créer une notification actionnée par un bouton dans une application
- Quand on clique sur le bouton, la notification apparaît en haut de l'écran
- Ajouter un bouton à la notification
- Lorsqu'on active ce bouton, on peut envoyer un toast par exemple



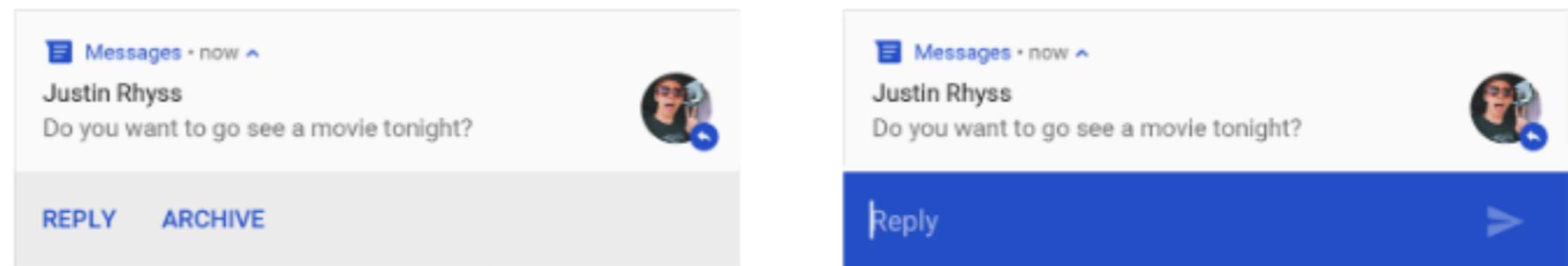
**Android**  
Studio

# SOLUTION PROPOSÉE

[HTTPS://DEVELOPER.ANDROID.COM/TRAINING/NOTIFY-USER/BUILD-NOTIFICATION](https://developer.android.com/training/notify-user/build-notification)

- Notifications
  - Overview
  - Create a notification**
  - Create an expandable notification
  - Start an activity from a notification
  - Create a group of notifications
  - Create and manage notification channels
  - Modify a notification Badge
  - Create a custom notification
- ▶ Add the app bar
- ▶ Control the system UI visibility
- ▶ Designing effective navigation
  - Implementing effective navigation
- ▶ Slide between fragments using ViewPager
- ▶ Supporting swipe-to-refresh
- Toasts overview
- ▶ Pop-up messages overview
- Dialogs

The direct reply action, introduced in Android 7.0 (API level 24), allows users to enter text directly into the notification, which is delivered to your app without opening an activity. For example, you can use a direct reply action to let users reply to text messages or update task lists from within the notification.



**Figure 3.** Tapping the "Reply" button opens the text input

The direct reply action appears as an additional button in the notification that opens a text input. When the user finishes typing, the system attaches the text response to the intent you had specified for the notification action and sends the intent to your app.

## Add the reply button

To create a notification action that supports direct reply:

1. Create an instance of `RemoteInput.Builder` that you can add to your notification action. This class's constructor accepts a string that the system uses as the key for the

# Notification

La Notification est le message qui apparaît sur l'écran de notre appareil mobile lorsque on n'utilise pas l'application.



**Android**  
Studio

# NotificationCompat

- \* API NotificationCompat de la bibliothèque de support Android.
- \* Ces API vous permettent d'ajouter des fonctionnalités disponibles uniquement sur les versions plus récentes d'Android tout en offrant une compatibilité avec Android 4.0 (API de niveau 14).

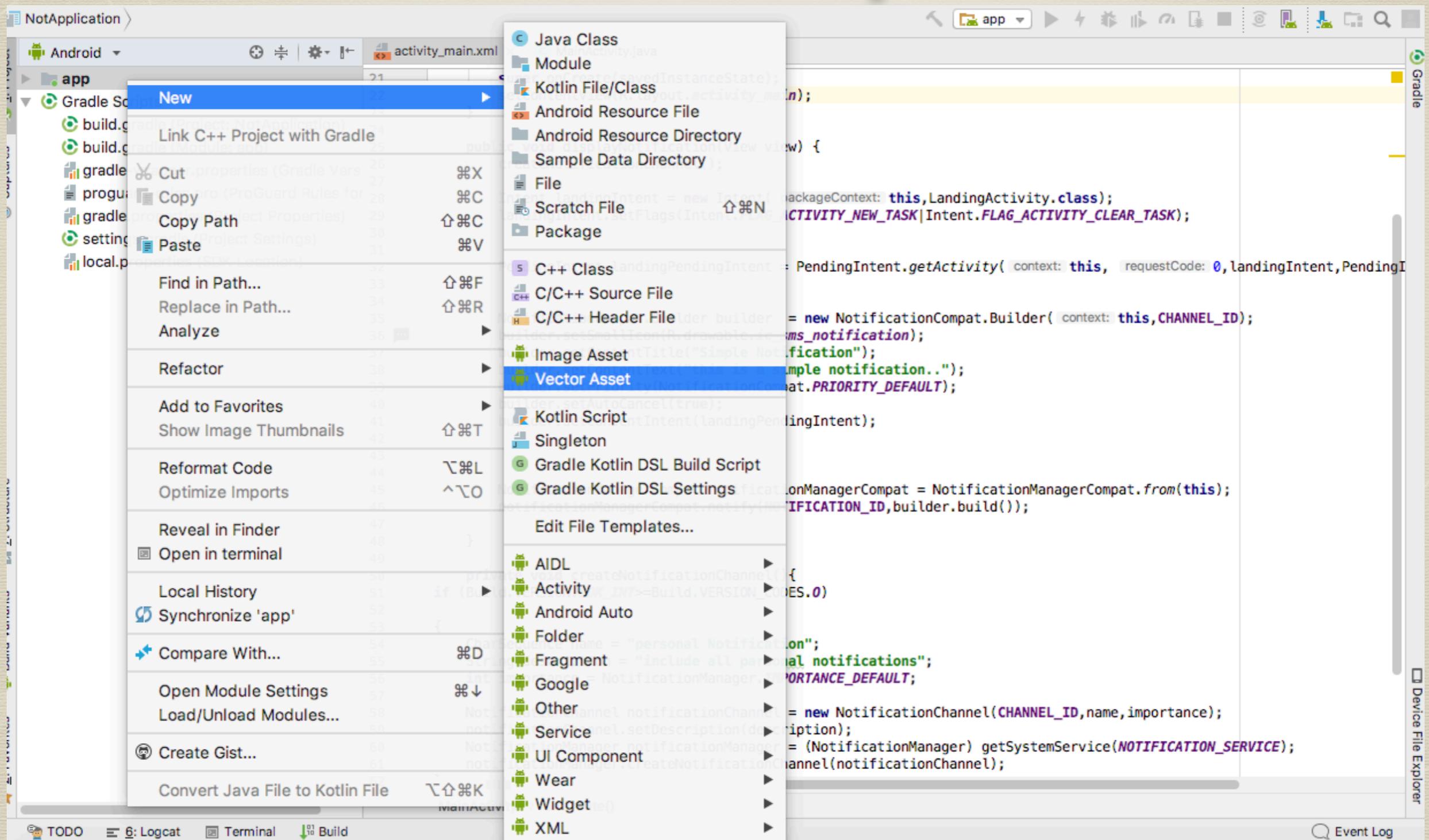


# CHANNEL\_ID

- \* Depuis Android 8.0 (API 26), il faut créer un canal de notification pour la création d'une notification
- \* Créez d'abord deux attributs :
- \* Un String CHANNEL\_ID correspondant à l'identifiant du canal
- \* Un entier NOTIFICATION\_ID correspondant à l'identifiant de la notification



# Les étapes de la création d'une notification actionnée par un bouton



The screenshot shows an IDE interface with the 'New' menu open. The 'Vector Asset' option is highlighted. The code editor in the background shows Kotlin code for creating a notification channel:

```
packageContext: this, LandingActivity.class);
ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);

PendingIntent.getActivity(context: this, requestCode: 0, landingIntent, PendingIntent

= new NotificationCompat.Builder(context: this, CHANNEL_ID);
ms_notification);
notification);
imple notification..");
at.PRIORITY_DEFAULT);

Intent);

NotificationManagerCompat = NotificationManagerCompat.from(this);
IFICATION_ID, builder.build());

{
ES.0)

on";
ial notifications";
ORTANCE_DEFAULT);

= new NotificationChannel(CHANNEL_ID, name, importance);
ription);
= (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
annel(notificationChannel);
```



# Configure Vector Asset

Android Studio

Asset Type:  Clip Art  Local file (SVG, PSD)

Name:

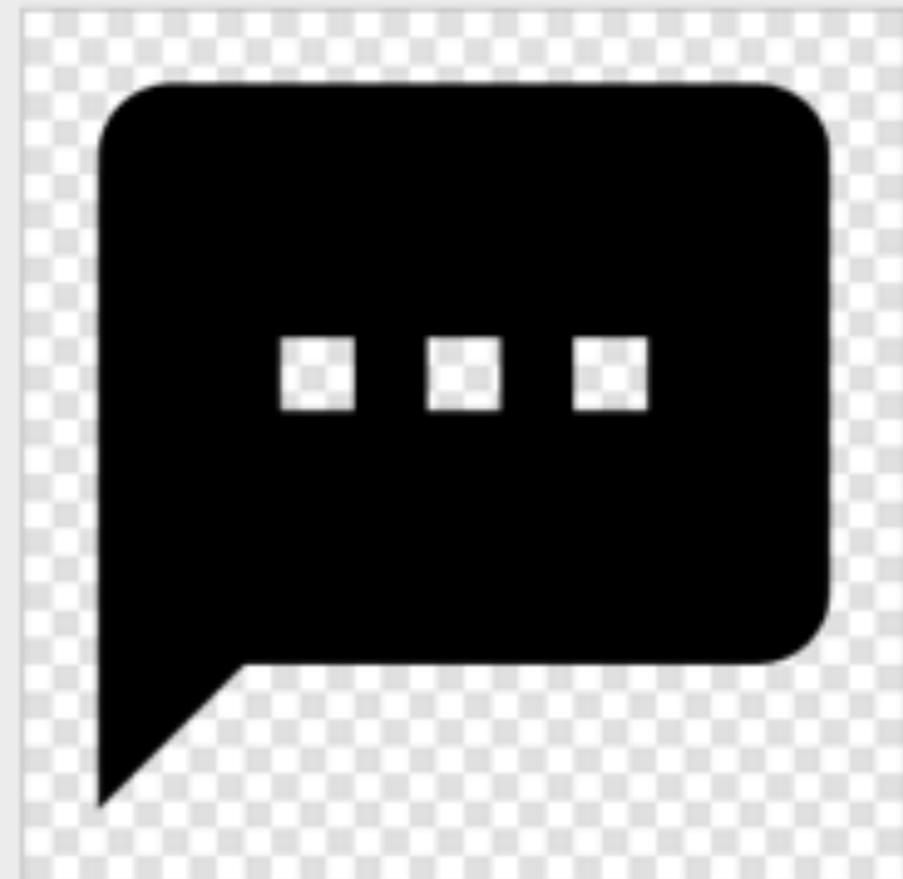
Clip Art: 

Size:  dp X  dp  Override

Color:

Opacity:  100 %

Enable auto mirroring for RTL layout



Vector Drawable Preview



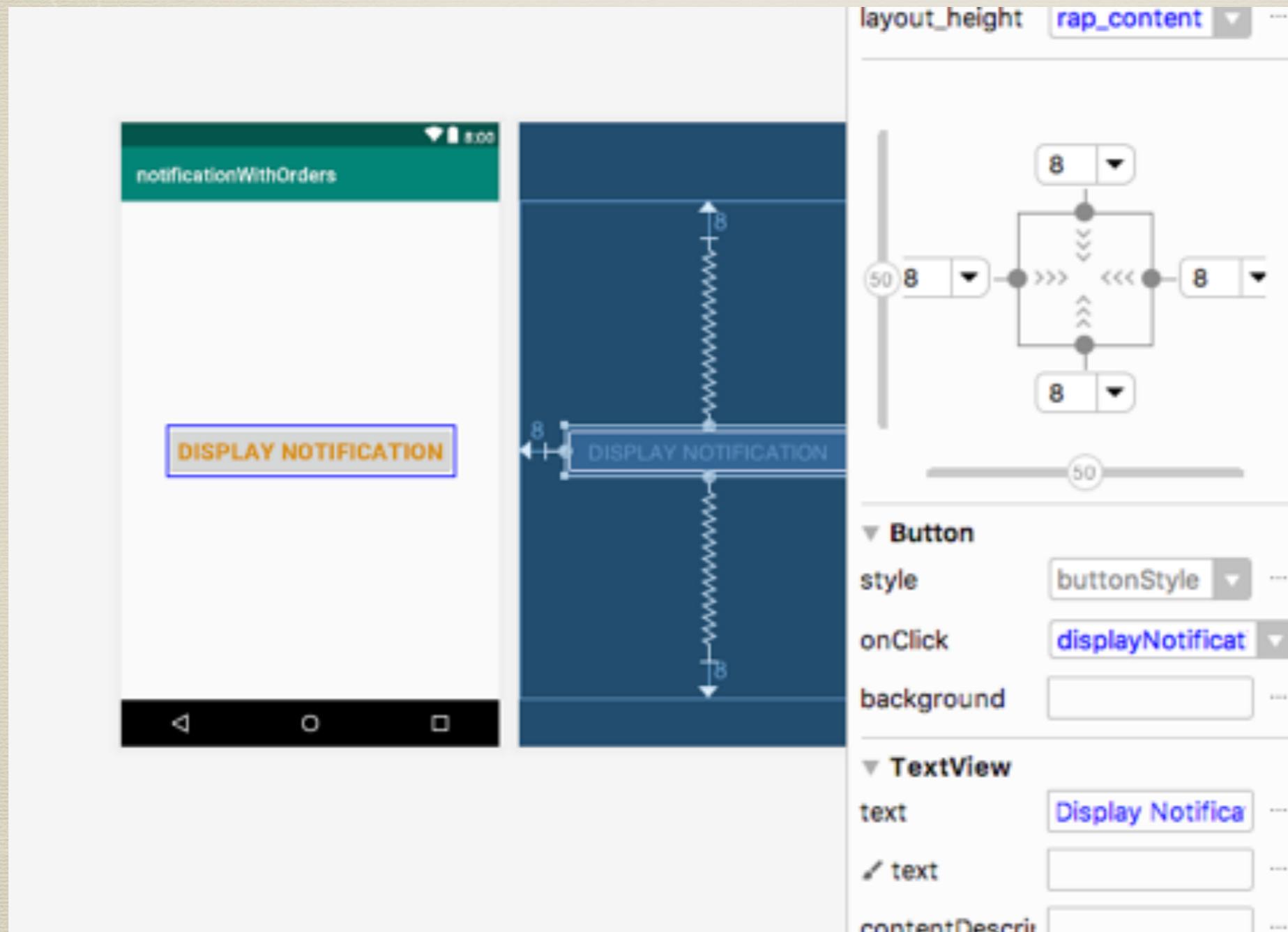
Cancel

Previous

Next

Finish

# Le bouton



# Code pour le bouton

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:onClick="displayNotification"
    android:text="Display Notification"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```



# Le code de la notification

```
public void displayNotification(View view) {  
    createNotificationChannel();  
  
    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, CHANNEL_ID);  
    builder.setSmallIcon(R.drawable.ic_sms_notification);  
    builder.setContentTitle("Simple Notification");  
    builder.setContentText("this is a simple notification..");  
    builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);  
    builder.setAutoCancel(true);  
  
    NotificationManagerCompat notificationManagerCompat = NotificationManagerCompat.from(this);  
    notificationManagerCompat.notify(NOTIFICATION_ID, builder.build());  
}  
  
private void createNotificationChannel(){  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O)  
    {  
        CharSequence name = "personal Notification";  
        String description = "include all personal notifications";  
        int importance = NotificationManager.IMPORTANCE_DEFAULT;  
  
        NotificationChannel notificationChannel = new NotificationChannel(CHANNEL_ID, name, importance);  
        notificationChannel.setDescription(description);  
        NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
        notificationManager.createNotificationChannel(notificationChannel);  
    }  
}
```

# Des boutons d'action

- \* Une notification peut offrir jusqu'à trois boutons d'action permettant à l'utilisateur de répondre rapidement, par exemple, répéter un rappel ou même répondre à un message texte. Mais ces boutons d'action ne doivent pas dupliquer l'action effectuée lorsque l'utilisateur appuie sur la notification.



# Ajouter un bouton d'action

- \* Pour ajouter un bouton d'action, passez un `PendingIntent` à `addAction ()`



# Ajouter une activity

The screenshot shows an IDE interface with the 'New' menu open. The 'Empty Activity' option is highlighted in blue. The background shows a project structure with folders like 'manifests', 'java', 'generatedJava', 'res', 'mipmap', and 'values'. The 'res' folder is expanded to show 'activity\_main.xml'. The 'MainActivity.java' file is open in the editor, showing code for a class that extends AppCompatActivity. The bottom status bar shows '17:1 LF UTF-8 Context: <no context>'.

**New**

- Link C++ Project with Gradle
- Cut
- Copy
- Copy Path
- Paste
- Find in Path...
- Replace in Path...
- Analyze
- Refactor
- Add to Favorites
- Show Image Thumbnails
- Reformat Code
- Optimize Imports
- Reveal in Finder
- Open in terminal
- Local History
- Synchronize 'app'
- Compare With...
- Open Module Settings
- Load/Unload Modules...
- Create Gist...
- Convert Java File to Kotlin File

**Java Class**

- Module
- Kotlin File/Class
- Android Resource File
- Android Resource Directory
- Sample Data Directory
- File
- Scratch File
- Package
- C++ Class
- C/C++ Source File
- C/C++ Header File
- Image Asset
- Vector Asset
- Kotlin Script
- Singleton
- Gradle Kotlin DSL Build Script
- Gradle Kotlin DSL Settings
- Edit File Templates...
- AIDL
- Activity
- Android Auto
- Folder
- Fragment
- Google
- Other
- Service
- UI Component
- Wear
- Widget
- XML
- Resource Bundle

**Gallery...**

- Android TV Activity (Requires minSdk >= 21)
- Android Things Empty Activity (Requires minSdk >= 24)
- Android Things Peripheral Activity (Requires minSdk >= 24)
- Basic Activity
- Blank Wear Activity (Requires minSdk >= 23)
- Bottom Navigation Activity
- Empty Activity
- Fragment + ViewModel
- Fullscreen Activity
- Login Activity
- Master/Detail Flow
- Navigation Drawer Activity
- Scrolling Activity
- Settings Activity
- Tabbed Activity

```
public class MainActivity extends AppCompatActivity {  
    private final Intent intent = new Intent(this, NotificationManager.class);  
    private final NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
    private final NotificationChannel notificationChannel = new NotificationChannel(CHANNEL_ID, name, importance);  
    private final Notification notification = new Notification(notificationManager, notificationChannel, notificationChannel);  
    private void createNotification() {  
        notificationManager.createNotificationChannel(notificationChannel);  
        notificationManager.notify(notificationChannel.getId(), notification);  
    }  
}
```

# Le code pour bouton de réponse

```
createNotificationChannel();

Intent landingIntent = new Intent( packageContext: this, LandingActivity.class);
landingIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);

PendingIntent landingPendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, landingIntent, PendingIntent.FLAG_ONE_SHOT);

Intent YesIntent = new Intent( packageContext: this, YesActivity.class);
YesIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent YesPendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, YesIntent, PendingIntent.FLAG_ONE_SHOT);

NotificationCompat.Builder builder = new NotificationCompat.Builder( context: this, CHANNEL_ID);
builder.setSmallIcon(R.drawable.ic_sms_notification);
builder.setContentTitle("Simple Notification");
builder.setContentText("this is a simple notification..");
builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);
builder.setAutoCancel(true);
builder.setContentIntent(landingPendingIntent);
builder.addAction(R.drawable.ic_sms_notification, title: "Yes", YesPendingIntent);

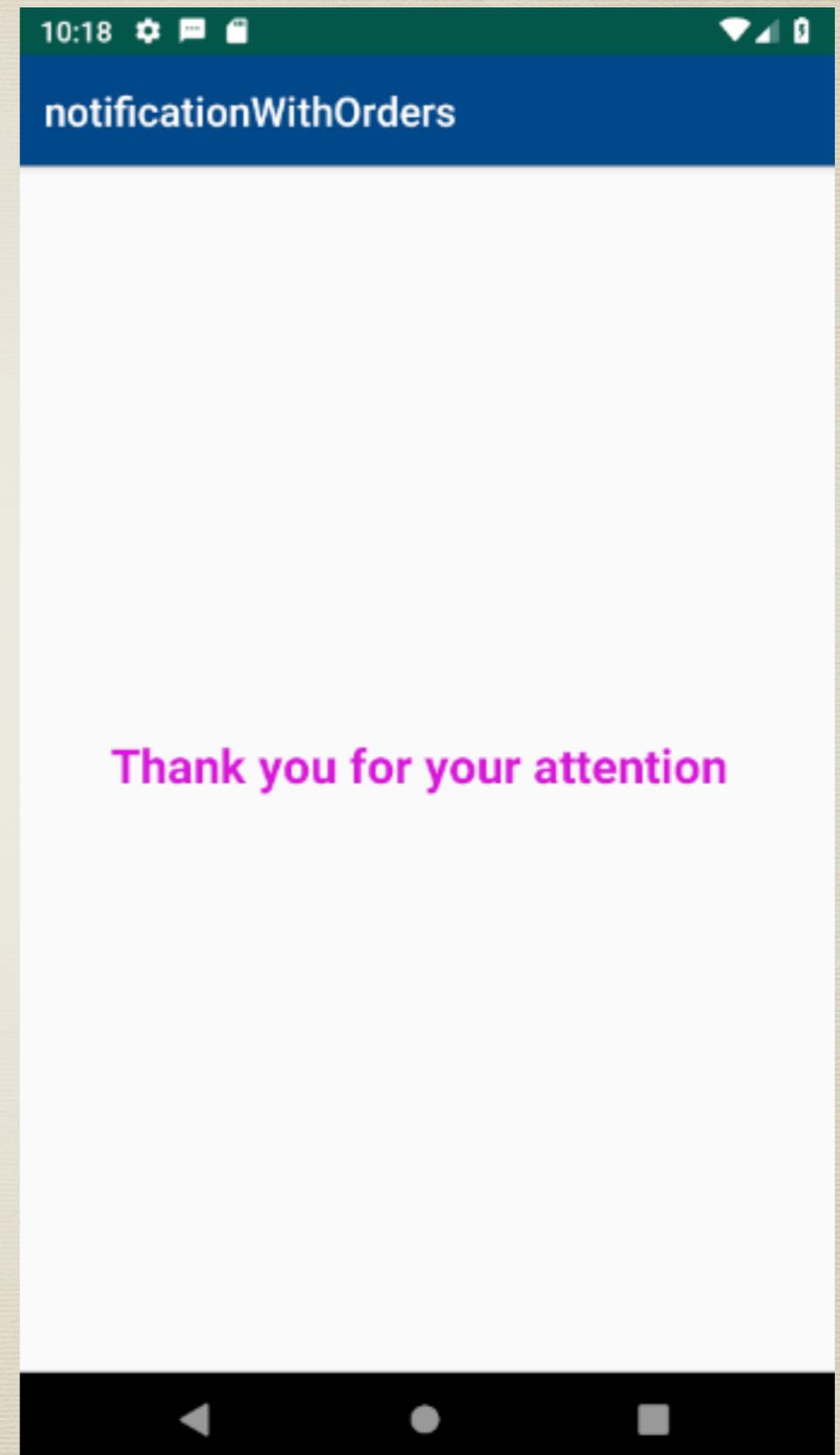
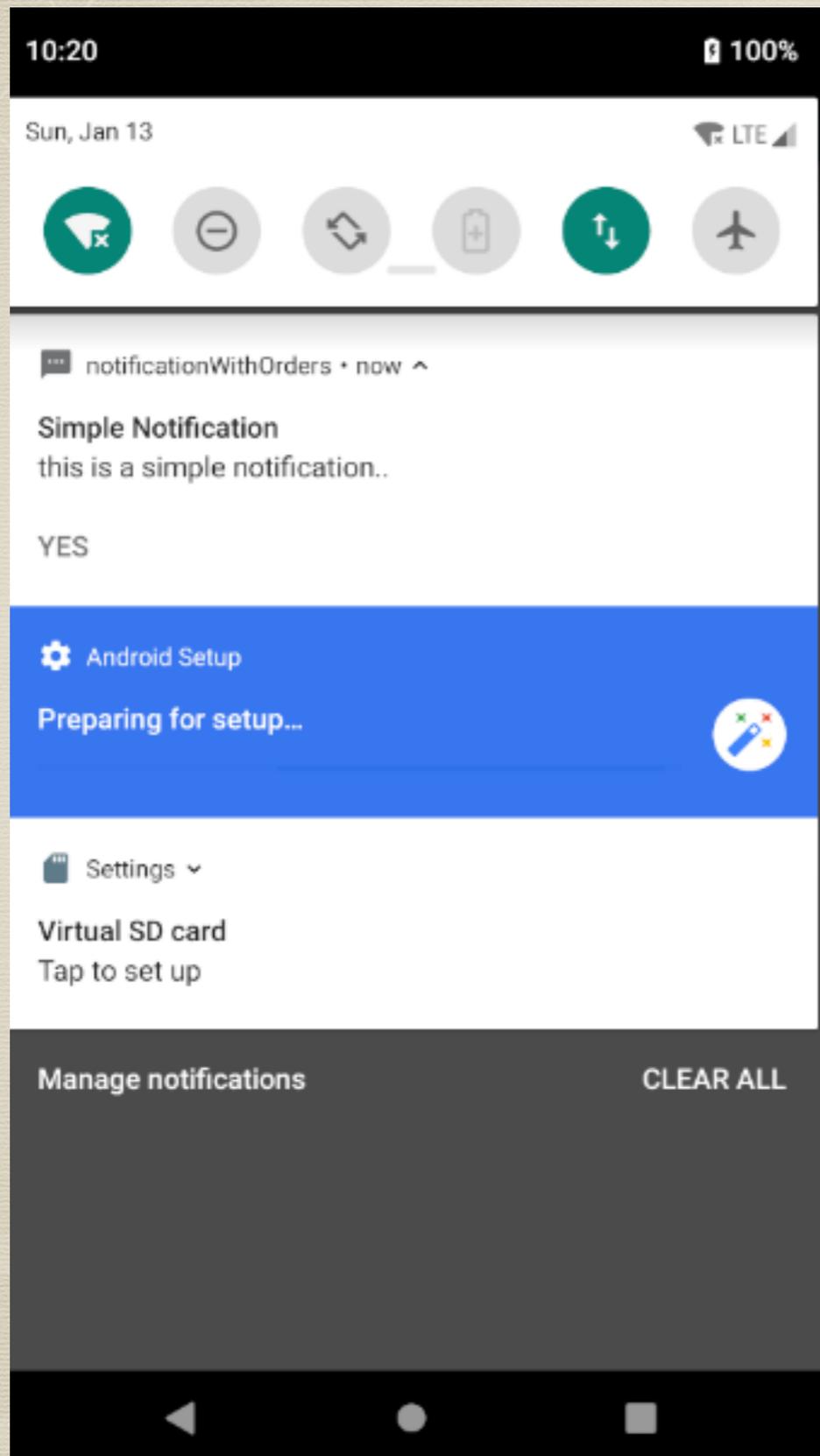
NotificationManagerCompat notificationManagerCompat = NotificationManagerCompat.from(this);
notificationManagerCompat.notify(NOTIFICATION_ID, builder.build());
}
```

# Texte View

The screenshot displays the Android Studio IDE with the following components:

- Project Explorer (Left):** Shows the project structure for 'app', including folders for manifests, java, generatedJava, res (drawable, layout), mipmap, values, and Gradle Scripts. The 'layout' folder contains activity\_landing.xml, activity\_main.xml, and activity\_yes2.xml.
- Palette (Top Left):** Lists widget categories: Common, Text, Buttons, Widgets, Layouts, Containers, Google, and Legacy. The 'Text' category is selected, showing 'Ab TextView' with sub-items: Button, ImageView, RecyclerView, <fragment>, ScrollView, and Switch.
- Component Tree (Bottom Left):** Shows a 'ConstraintLayout' containing an 'Ab textView2' with the text 'Thank y...' and a warning icon.
- Design View (Center):** Shows a mobile device simulation with a dark blue header 'notificationWithOrders' and a pink text view 'Thank you for your attention'. A blue box highlights the text view with dimension lines indicating 8dp margins on all sides.
- Attributes (Right):** An empty panel for widget attributes.
- Toolbar (Top):** Includes navigation icons, device selection (Nexus 4), API level (28), zoom (24%), and other tool icons.

# ÇA DONNE QUOI EN VRAI ?



# Sources

[HTTPS://DEVELOPER.ANDROID.COM/TRAINING/NOTIFY-USER/BUILD-NOTIFICATION](https://developer.android.com/training/notify-user/build-notification)

- Vous retrouverez ce lien dans : Android : Sujets de mini-projets individuels
- Une question ? Envoyez moi un mail : [souad.bellouni@gmail.com](mailto:souad.bellouni@gmail.com)

Merci de votre attention



"La connaissance s'acquiert  
par l'expérience, tout le reste  
n'est que de l'informa