

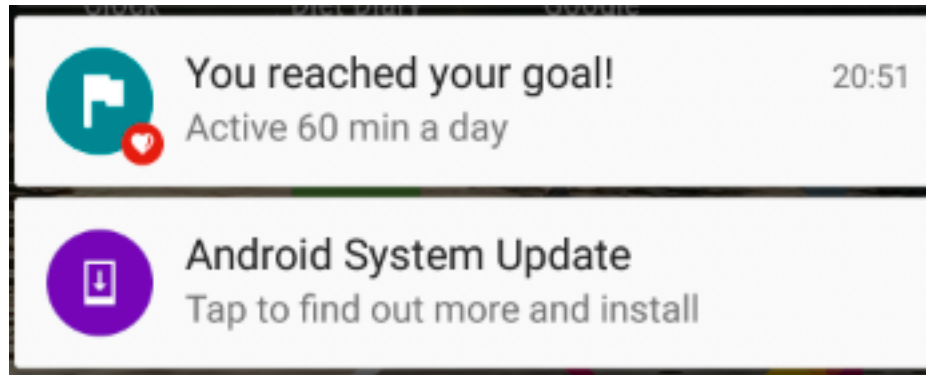
NOTIFICATIONS

Qu'est-ce qu'une notification?

- Une **notification** est un court message affiché brièvement sur la zone d'affichage des notifications
- Il annonce généralement la survenue d'un événement spécial pour lequel un déclencheur a été défini.
- Après avoir ouvert le panneau de notification, l'utilisateur peut choisir de cliquer sur une sélection et d'exécuter une activité associée.

Qu'est-ce qu'une notification?

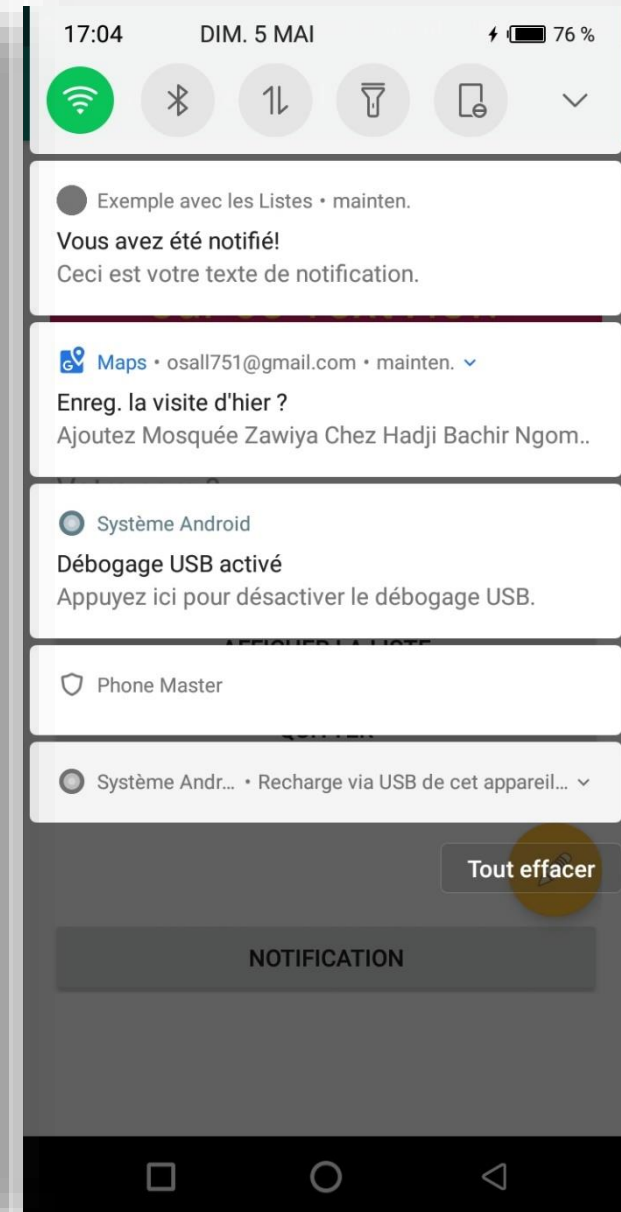
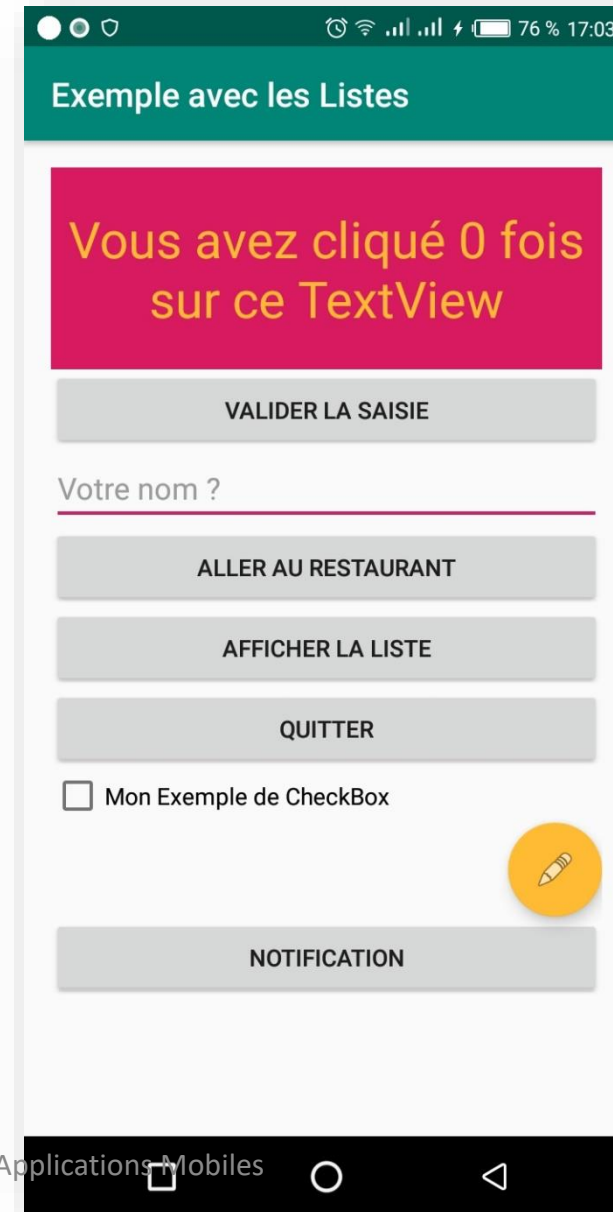
Message affiché à l'utilisateur en dehors de l'interface utilisateur de l'application normale



- Small icon
- Title
- Detail text

Comment les notifications sont-elles utilisées?

- Android émet une notification qui apparaît sous forme d'icône dans la barre d'état.
- Pour voir les détails, l'utilisateur ouvre le tiroir de notification.
- L'utilisateur peut afficher les notifications à tout moment dans le tiroir de notifications.



Notifications

NotificationManager

Cette classe informe l'utilisateur des événements qui se produisent en arrière-plan. Les notifications peuvent prendre différentes formes:

1. Une icône persistante qui va dans la barre d'état et est accessible via le lanceur, (lorsque l'utilisateur la sélectionne, une intention désignée peut être lancée),
2. allumer ou clignoter les voyants de l'appareil, ou
3. Alerte à l'utilisateur en faisant clignoter le rétroéclairage, en jouant un son ou en faisant vibrer.

Notifications

NotificationManager

Vous n'instanciez pas directement cette classe; à la place, récupérez-la via **getSystemService (String)**.

Example:

```
public void afficherNotification(String titre, String message) {  
    String nomService = Context.NOTIFICATION_SERVICE;  
    notificationManager = (NotificationManager) getSystemService(nomService);  
}
```

Notifications

NotificationManager

Vous n'instanciez pas directement cette classe; à la place, récupérez-la via **getSystemService (String)**.

Example:

```
public void afficherNotification(String titre, String message) {  
    String nomService = Context.NOTIFICATION_SERVICE;  
    notificationManager = (NotificationManager) getSystemService(nomService);  
}
```

Créer une notification

Notification.Builder

- La notification est créée à l'aide de la classe **NotificationCompat.Builder**.
- Transmettez le contexte de l'application et l'ID du canal de notification au constructeur.
- Le constructeur **NotificationCompat.Builder** utilise l'ID du canal de notification. Ce dernier n'est utilisé que par Android 8.0 (API de niveau 26) et supérieur, mais ce paramètre est ignoré par les versions antérieures.

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this, CHANNEL_ID)
```


Notification channels

- Utilisé pour créer un canal personnalisable par l'utilisateur pour chaque type de notification à afficher.
- Plusieurs notifications peuvent être regroupées dans un canal.
- Définissez le comportement de notification comme son, lumière, vibration, etc., appliqué à toutes les notifications de ce canal.

Notification android: méthodes

```
m canBubble() boolean
m canBypassDnd() boolean
m canShowBadge() boolean
m describeContents() int
m enableLights(boolean lights) void
m enableVibration(boolean vibration) void
m equals(Object o) boolean
m getAudioAttributes() AudioAttributes
m getConversationId() String
m getDescription() String
m getGroup() String
m getId() String
m getImportance() int
m getLightColor() int
m getLockscreenVisibility() int
m getName() CharSequence
m getParentChannelId() String
m getSound() Uri
m getVibrationPattern() long[]
m hashCode() int
m hasUserSetImportance() boolean
m hasUserSetSound() boolean
```

Notification android: méthodes

```
m isImportantConversation() boolean
m setAllowBubbles(boolean allowBubbles) void
m setBypassDnd(boolean bypassDnd) void
m setConversationId(String channelId, String conversationId) void
m setDescription(String description) void
m setGroup(String groupId) void
m setImportance(int importance) void
m setLightColor(int argb) void
m setLockscreenVisibility(int lockscreenVisibility) void
m setName(CharSequence name) void
m setShowBadge(boolean showBadge) void
m setSound(Uri sound, AudioAttributes audioAttributes) void
m setVibrationPattern(long[] vibrationPattern) void
m shouldShowLights() boolean
m shouldVibrate() boolean
m toString() String
m writeToParcel(Parcel dest, int flags) void
m getClass() Class<? extends NotificationChannel>
m notify() void
m notifyAll() void
m wait() void
m wait(long timeout) void
m wait(long l, int i) void
```

Les canaux de notification sont obligatoires

- Les canaux de notification sont introduits dans Android 8.0 (API niveau 26)
- Toutes les notifications doivent être attribuées à un canal à partir d'Android 8.0 (API niveau 26), sinon vos notifications ne seront pas affichées.
- Pour les applications ciblant une version inférieure à Android 8.0 (API niveau 26), nul besoin d'implémenter des canaux de notification.

Les canaux de notification sont obligatoires

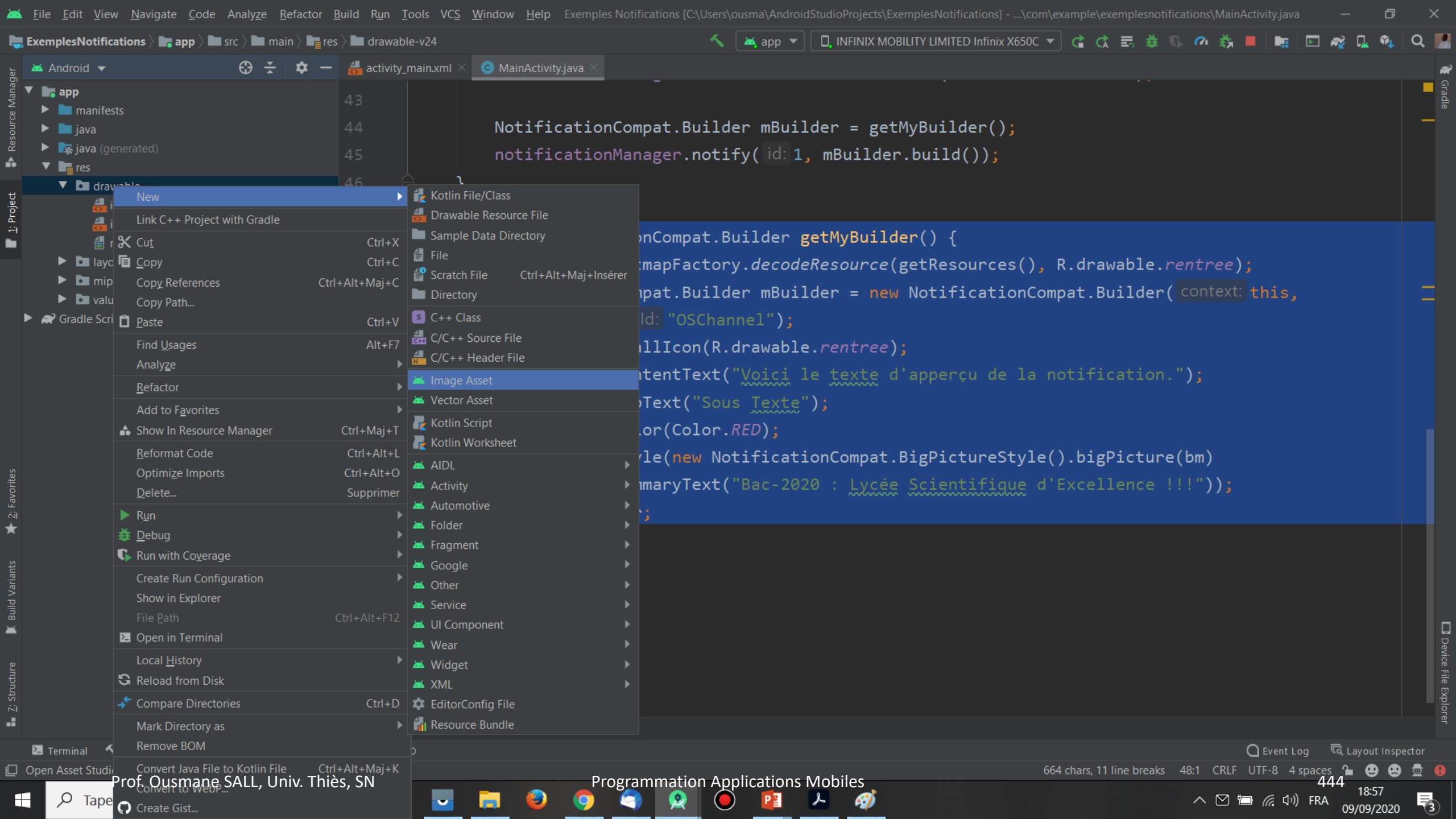
```
@RequiresApi(api = Build.VERSION_CODES.O)
public void afficherNotification(String titre, String message) {
    String nomService = Context.NOTIFICATION_SERVICE;
    notificationManager = (NotificationManager) getSystemService(nomService);

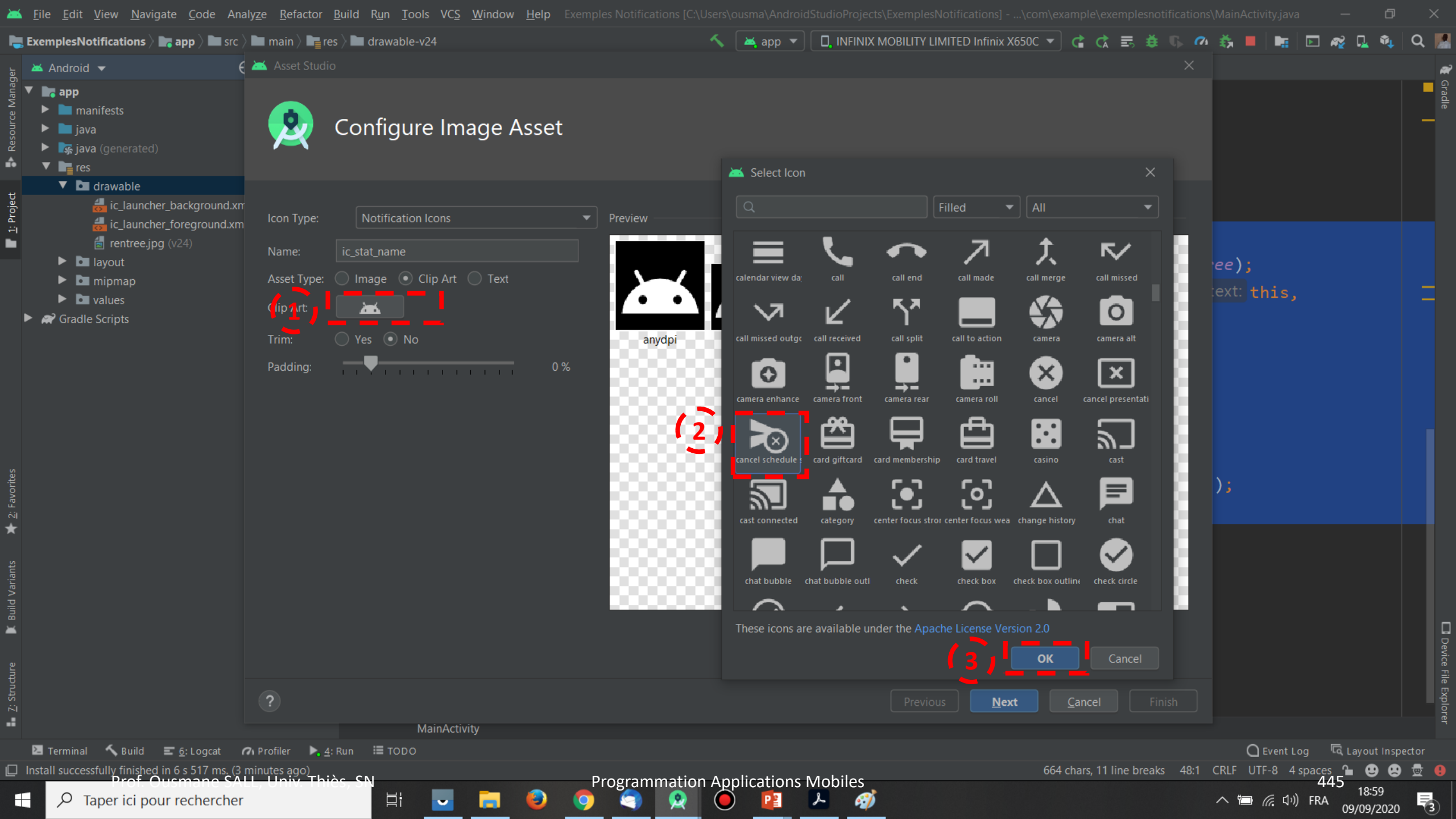
    NotificationChannel notificationChannel = new NotificationChannel("OSChannel",
        "Notification de Ousmane SALL", NotificationManager.IMPORTANCE_DEFAULT);
    notificationChannel.enableLights(true);
    notificationChannel.setLightColor(Color.RED);
    notificationChannel.enableVibration(true);
    notificationChannel.setDescription("Notification L3");
    notificationChannel.canShowBadge();
    notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);
    notificationManager.createNotificationChannel(notificationChannel);
}
```

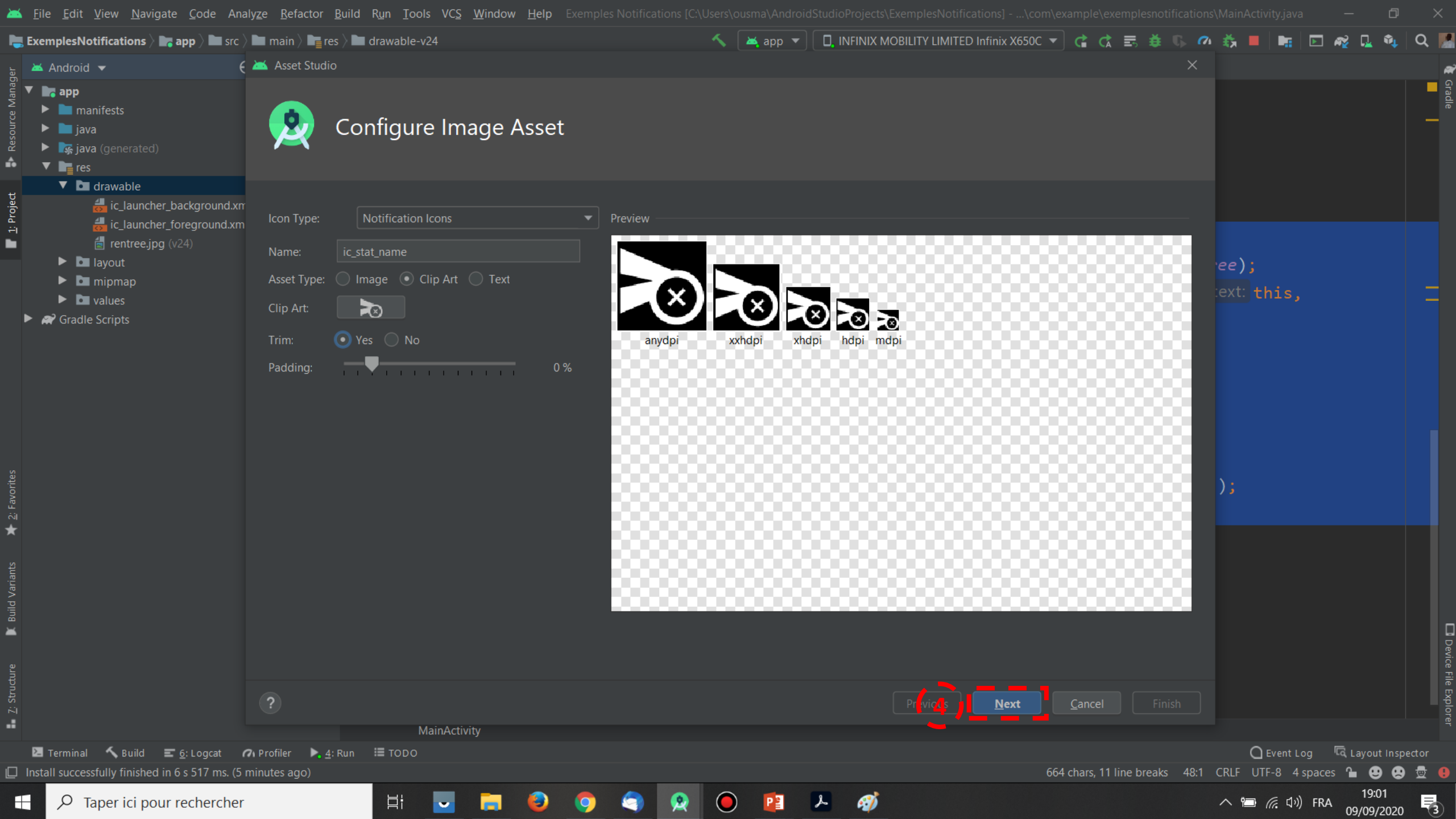
Créer un objet Notification

- **Notification.Builder** fournit une interface de générateur pour créer un objet Notification comme indiqué ci-dessous
- Transmettez le contexte de l'application et l'ID du canal de notification au constructeur.
- Le constructeur **NotificationCompat.Builder** utilise l'ID du canal de notification. Ce dernier n'est utilisé que par Android 8.0 (API de niveau 26) et supérieur, mais ce paramètre est ignoré par les versions antérieures.

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this, CHANNEL_ID);
```





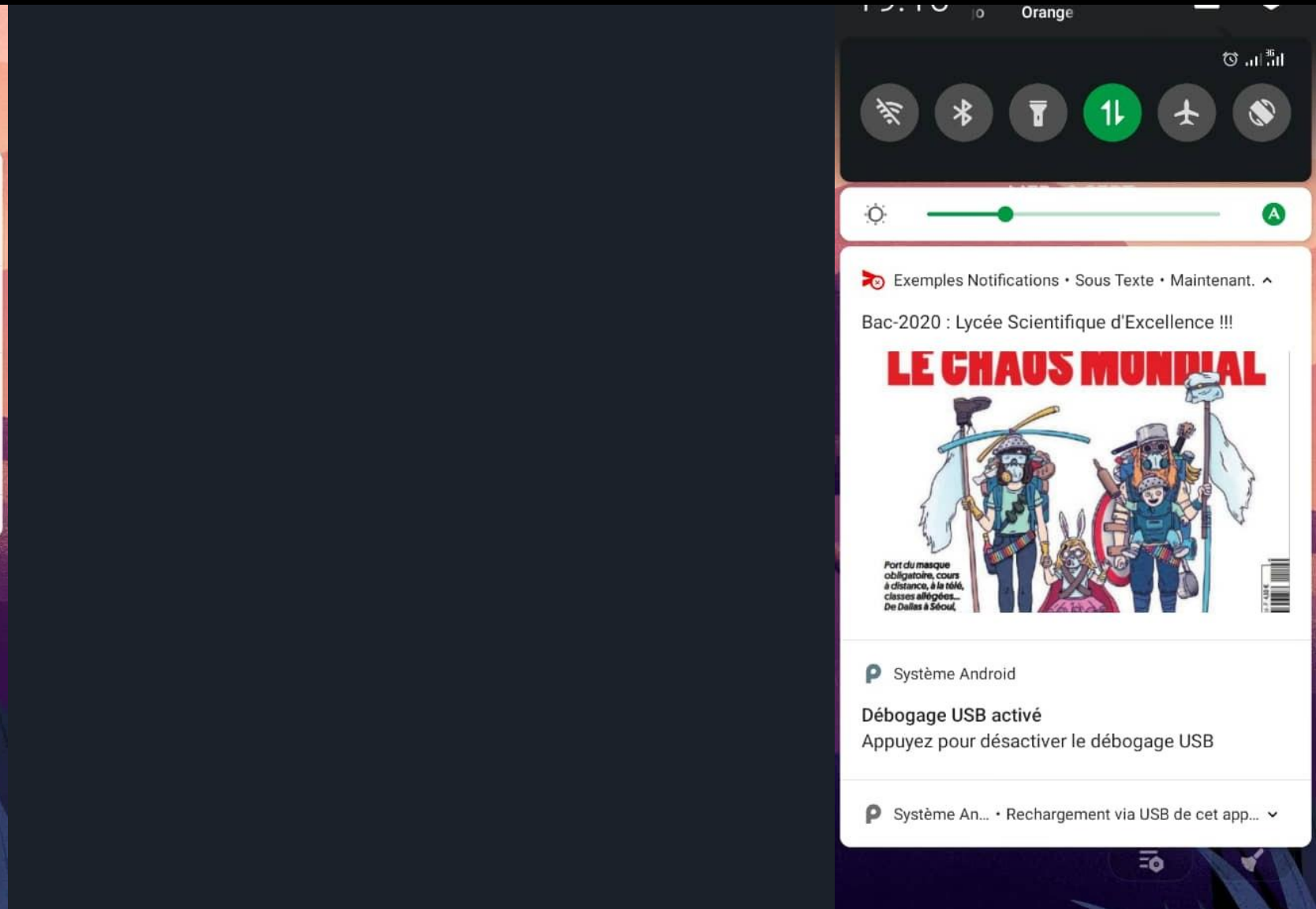
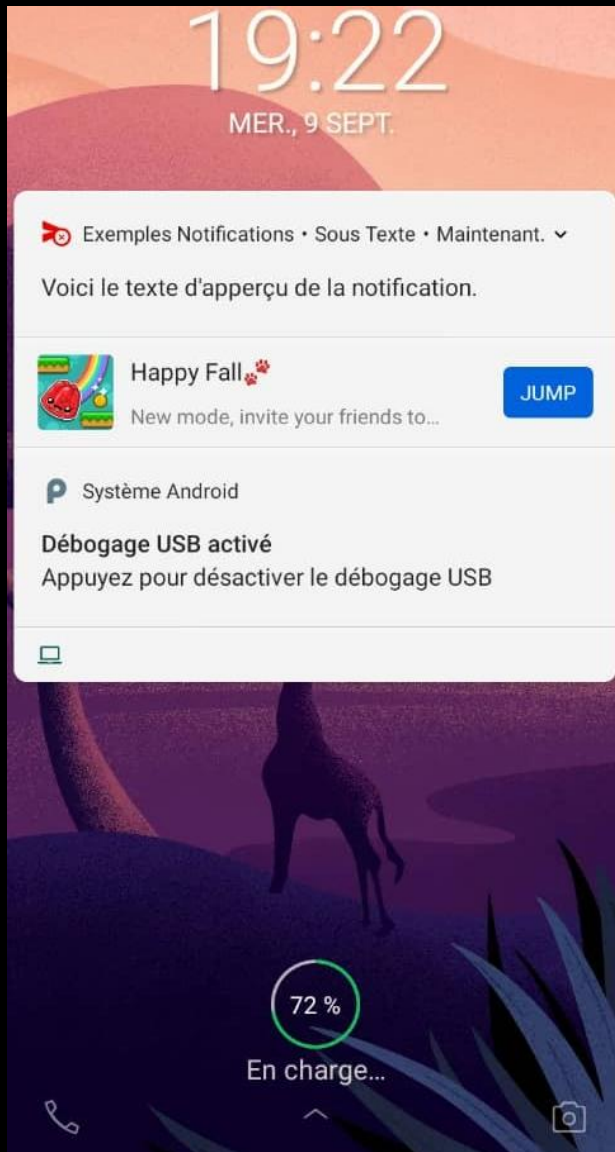


Créer un objet Notification

```
private NotificationCompat.Builder getMyBuilder() {  
    Bitmap bm = BitmapFactory.decodeResource(getResources(), R.drawable.rentree);  
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this,  
        "OSChannel");  
    mBuilder.setSmallIcon(R.drawable.ic_stat_name);  
    mBuilder.setContentText("Voici le texte d'aperçu de la notification.");  
    mBuilder.setSubText("Sous Texte");  
    mBuilder.setColor(Color.RED);  
    mBuilder.setStyle(new NotificationCompat.BigPictureStyle().bigPicture(bm)  
        .setSummaryText("Bac-2020 : Lycée Scientifique d'Excellence !!!"));  
    return mBuilder;  
}
```

Les canaux de notification sont obligatoires

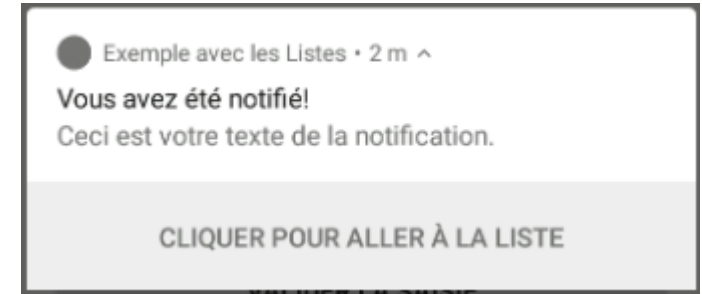
```
@RequiresApi(api = Build.VERSION_CODES.O)
public void afficherNotification(String titre, String message) {
    String nomService = Context.NOTIFICATION_SERVICE;
    notificationManager = (NotificationManager) getSystemService(nomService);
    NotificationChannel notificationChannel = new NotificationChannel("OSChannel",
        "Notification de Ousmane SALL", NotificationManager.IMPORTANCE_DEFAULT);
    notificationChannel.enableLights(true);
    notificationChannel.setLightColor(Color.RED);
    notificationChannel.enableVibration(true);
    notificationChannel.setDescription("Notification L3");
    notificationChannel.canShowBadge();
    notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);
    notificationManager.createNotificationChannel(notificationChannel);
    [NotificationCompat.Builder mBuilder = getMyBuilder();]
    notificationManager.notify(1, mBuilder.build());
}
```



Ajouter une notification

- Chaque notification doit répondre lorsqu'elle est exploitée, généralement en lançant une activité dans votre application.
- Définissez une intention de contenu à l'aide de la méthode `setContentIntent ()`.
- Passez l'intention emballée dans un objet `PendingIntent`.

Notification action buttons



- Les boutons d'action peuvent effectuer diverses actions pour le compte de votre application, telles que démarrer une tâche en arrière-plan, passer un appel téléphonique, etc.
- À partir d'Android 7.0 (API niveau 24), répondez aux messages directement à partir des notifications.
- Pour ajouter un bouton d'action, transmettez un PendingIntent à la méthode **addAction ()**.

Pending intents

- Un PendingIntent est une description d'une intention et d'une action cible à exécuter avec.
- Attribuez un objet PendingIntent à une autre application pour lui donner le droit d'effectuer l'opération que vous avez spécifiée comme si l'autre application était vous-même.

Méthodes pour créer un PendingIntent

- Pour instancier un PendingIntent, appliquez l'une des méthodes suivantes:
 - `PendingIntent.getActivity ()`
 - `PendingIntent.getBroadcast ()`
 - `PendingIntent.getService ()`

Arguments de la méthode PendingIntent

1. Contexte d'application
2. Code de demande - ID entier constant pour l'intention en attente
3. Intention d'être délivré
4. L'indicateur PendingIntent détermine comment le système gère plusieurs intentions en attente provenant de la même application.

Etape 1: Créer un Intent

```
Intent notificationIntent = new Intent(Intent.ACTION_VIEW,  
    Uri.parse("https://www.seneweb.com/news/Education/bac-2020-1e-lycee-scientifique-d-excelle_n_328262.html"));
```

Etape 2: Créer une instance de PendingIntent

```
PendingIntent notificationPendingIntent = PendingIntent.getActivity(  
    this,  
    0,  
    notificationIntent,  
    PendingIntent.FLAG_UPDATE_CURRENT);
```

Etape 3: Ajouter au notification builder

```
mBuilder.addAction(android.R.drawable.ic_menu_view, "Ouvrir", notificationPendingIntent);
```

Créer l'intent pour aller à Seneweb

```
Intent notificationIntent = new Intent(Intent.ACTION_VIEW,  
    Uri.parse("https://www.seneweb.com/news/Education/bac-2020-le-lycee-scientifique-d-excelle_n_328262.html"));  
PendingIntent notificationPendingIntent = PendingIntent.getActivity(  
    context: this,  
    requestCode: 0,  
    notificationIntent,  
    PendingIntent.FLAG_UPDATE_CURRENT);  
mBuilder.addAction(android.R.drawable.ic_menu_view, title: "Ouvrir", notificationPendingIntent);
```

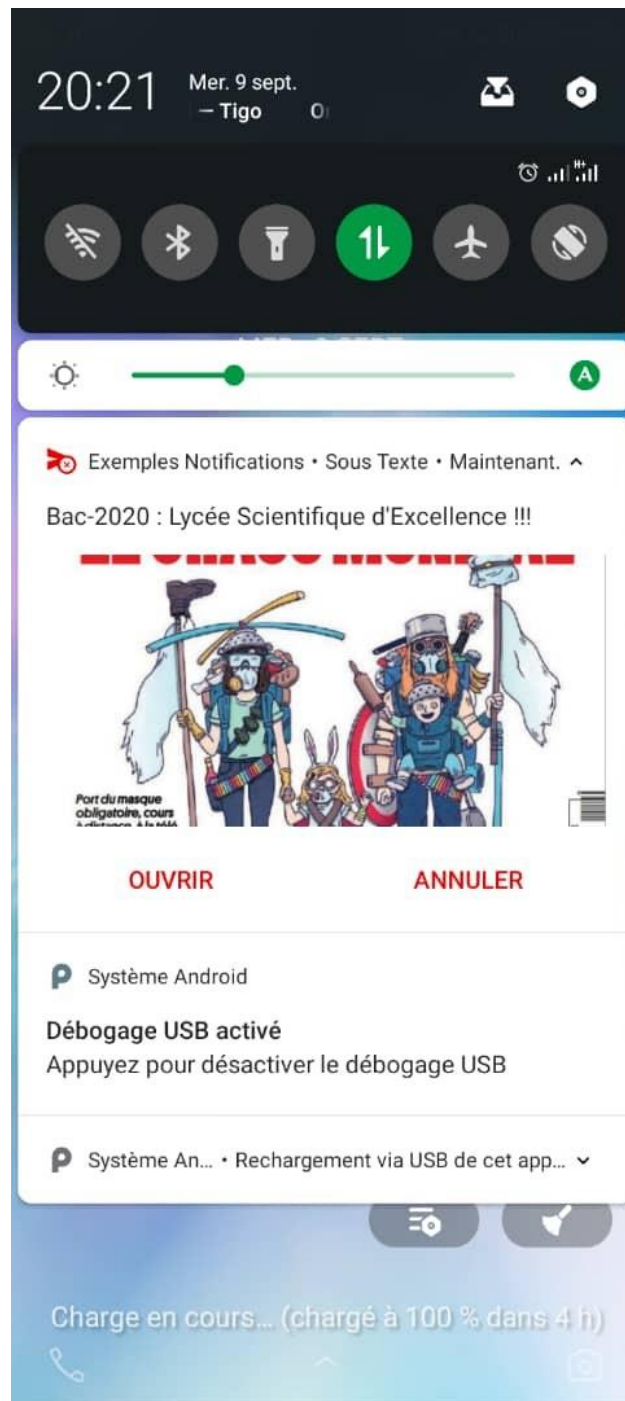
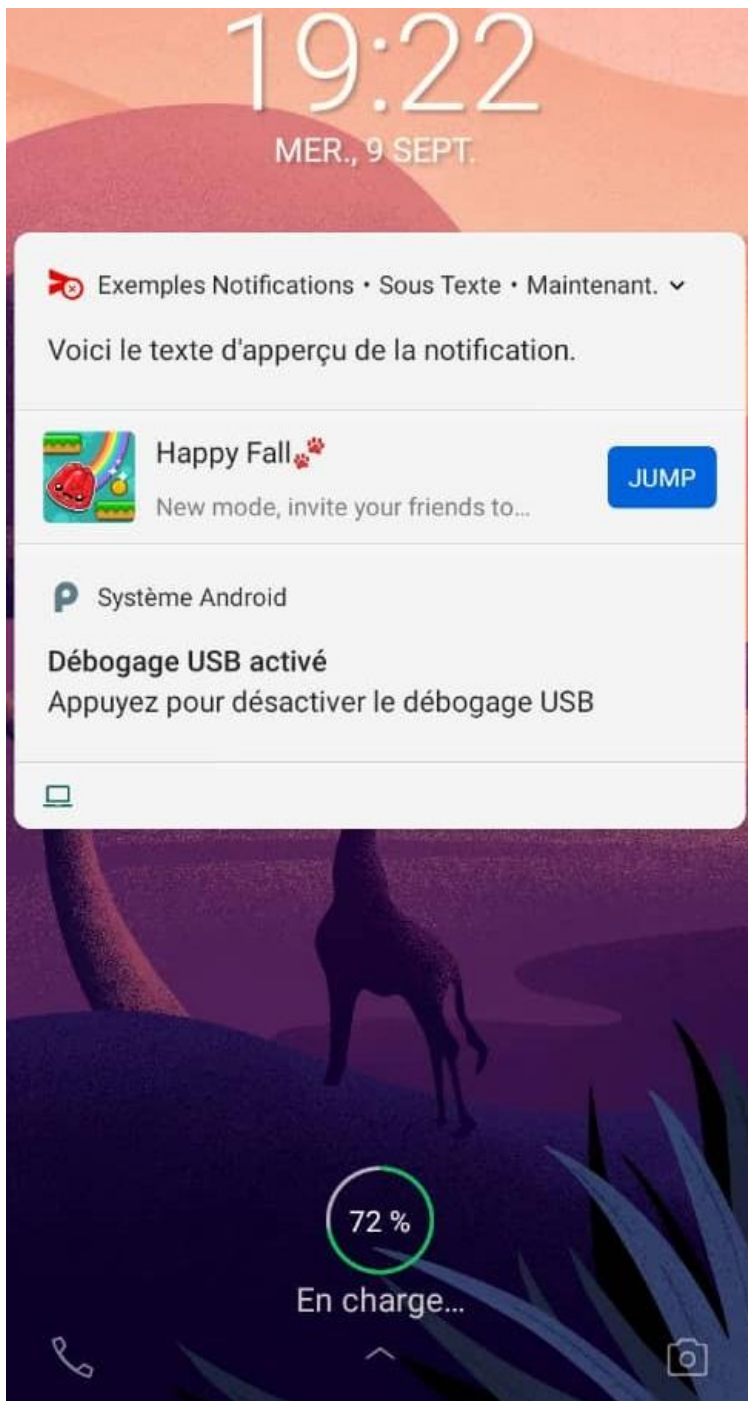
Créer l'Intent pour retourner à l'application

```
Intent annulerIntent = new Intent(this, MainActivity.class);
PendingIntent annulerPendingIntent = PendingIntent.getActivity(
    this,
    0,
    annulerIntent,
    PendingIntent.FLAG_CANCEL_CURRENT);
mBuilder.addAction(android.R.drawable.ic_delete, "Annuler", annulerPendingIntent);
```

```

private NotificationCompat.Builder getMyBuilder() {
    Bitmap bm = BitmapFactory.decodeResource(getResources(), R.drawable.rentree);
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(context: this,
        channelId: "OSChannel");
    mBuilder.setSmallIcon(R.drawable.ic_stat_name);
    mBuilder.setContentText("Voici le texte d'aperçu de la notification.");
    mBuilder.setSubText("Sous Texte");
    mBuilder.setColor(Color.RED);
    mBuilder.setStyle(new NotificationCompat.BigPictureStyle().bigPicture(bm)
        .setSummaryText("Bac-2020 : Lycée Scientifique d'Excellence !!!"));
    Intent notificationIntent = new Intent(Intent.ACTION_VIEW,
        Uri.parse("https://www.seneweb.com/news/Education/bac-2020-le-lycee-scientifique-d-excelle_n_328262.html"));
    PendingIntent notificationPendingIntent = PendingIntent.getActivity(
        context: this,
        requestCode: 0,
        notificationIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);
    mBuilder.addAction(android.R.drawable.ic_menu_view, title: "Ouvrir", notificationPendingIntent);
    Intent annulerIntent = new Intent(packageContext: this, MainActivity.class);
    PendingIntent annulerPendingIntent = PendingIntent.getActivity(
        context: this,
        requestCode: 0,
        annulerIntent,
        PendingIntent.FLAG_CANCEL_CURRENT);
    mBuilder.addAction(android.R.drawable.ic_delete, title: "Annuler", annulerPendingIntent);
    return mBuilder;
}

```

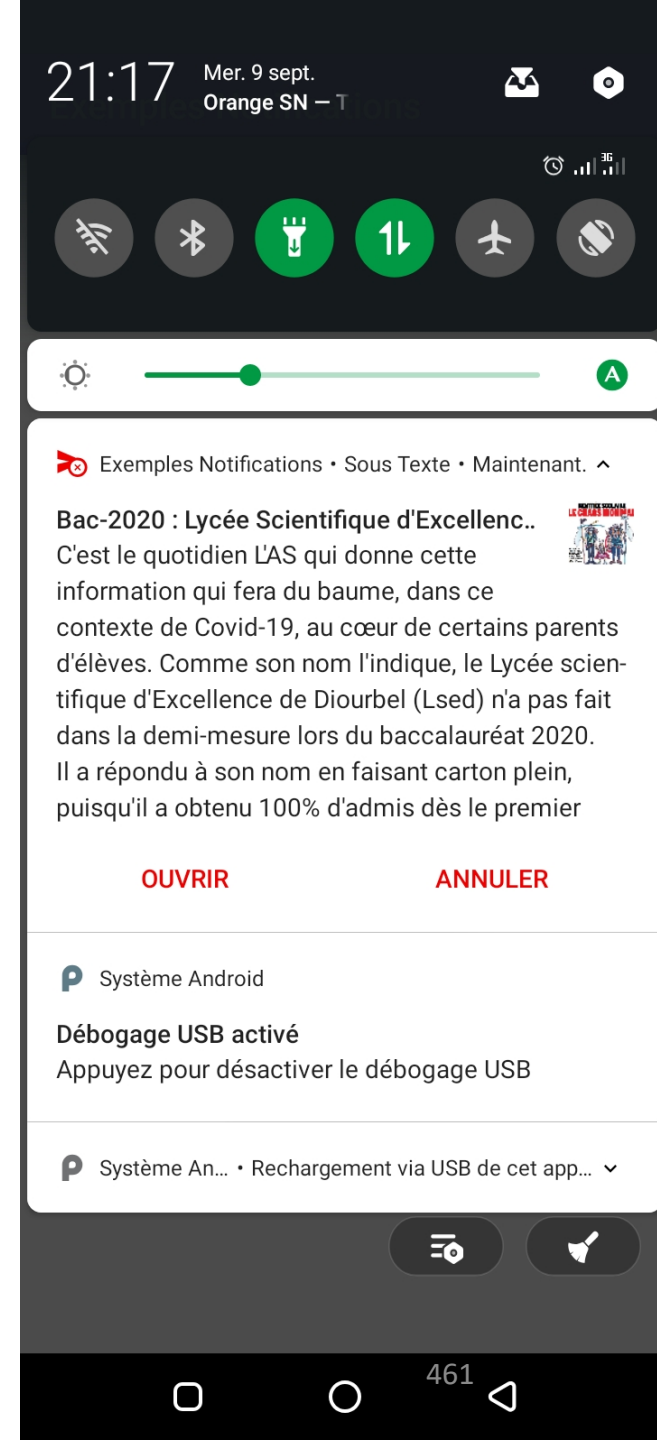


Style de notification

Android permet d'ajouter des styles riches à notre application avec l'introduction d'Android L. L'utilisation de ces styles rendrait la notification plus attrayante que jamais. Certains des styles connus utilisés dans de nombreuses applications sont répertoriés ci-dessous et sont explicites.

- **BigTextStyle**
- **BigPictureStyle**
- **InboxStyle**
- **MessageStyle**

BigTextStyle Notification

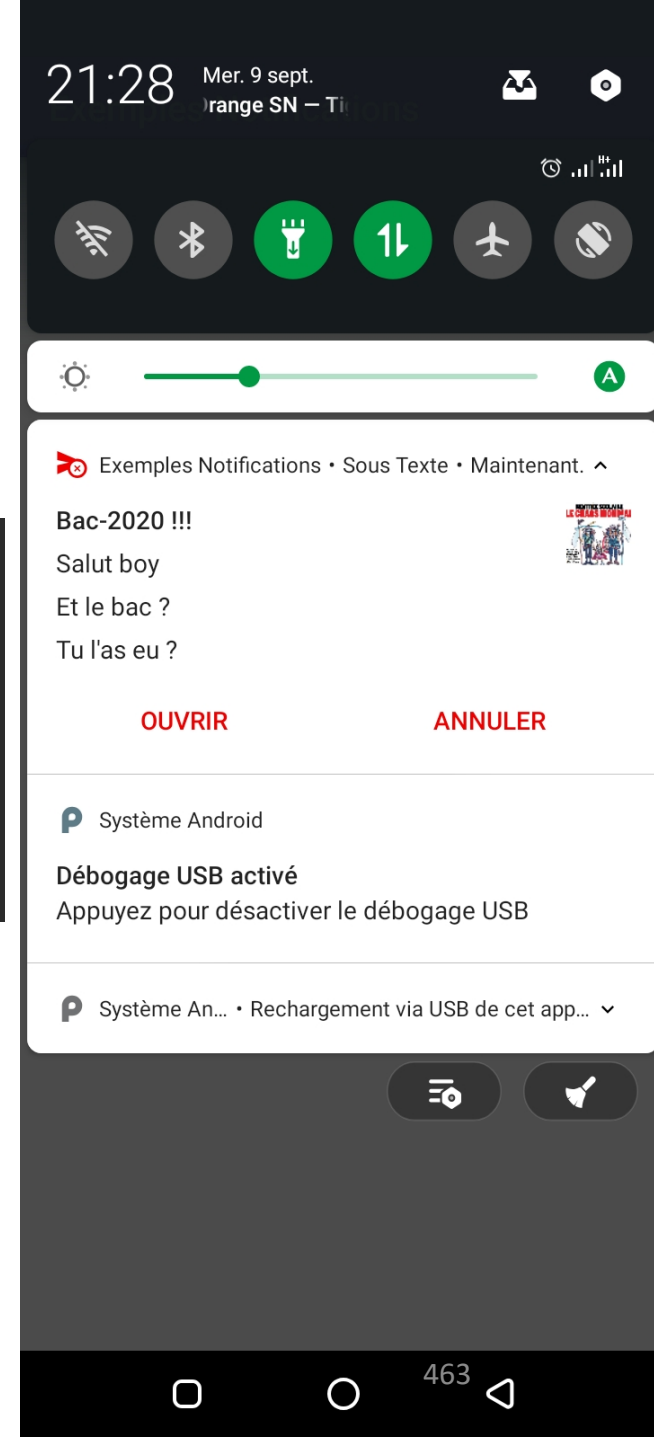


BigTextStyle Notification

```
mBuilder.setStyle(new NotificationCompat
    .BigTextStyle()
    .setBigContentTitle("Bac-2020 : Lycée Scientifique d'Excellence !!!")
    .bigText("C'est le quotidien L'AS qui donne cette information qui fera du baume, dans ce contexte de Covid-19, " +
        "au cœur de certains parents d'élèves. Comme son nom l'indique, le Lycée scientifique d'Excellence de Diourbel (Lsed) " +
        "n'a pas fait dans la demi-mesure lors du baccalauréat 2020. Il a répondu à son nom en faisant carton plein, puisqu'il a " +
        "obtenu 100% d'admis dès le premier tour. En Série S1, tous les 27 candidats qui se sont présentés ont été admis à l'issue" +
        " du premier groupe.")
    );
mBuilder.setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.rentree));
```

InboxStyle Notification

```
mBuilder.setStyle(new NotificationCompat
    .InboxStyle().addLine("Salut boy")
    .addLine("Et le bac ?")
    .addLine("Tu l'as eu ?")
    .setBigContentTitle("Bac-2020 !!!")
    .setSummaryText("Bac-2020 : Lycée Scientifique d'Excellence !!!")
);
```



Message Style Notification