



android

Система асинхронных оповещений пользователя

Выполнил студент группы М-РИТ-25-1
Станиславчук Сергей

Содержание

1. Уведомления

2. Тосты

Уведомления

Зачем они нужны и как их использовать

Уведомления. Зачем они нужны?

- Информируют пользователя о событиях, когда приложение не активно.
- Краткие, своевременные и релевантные сообщения.
- Контроль со стороны ОС и возможности разработчика влиять на внешний вид и поведение.

Уведомления. Шаги для проектирования

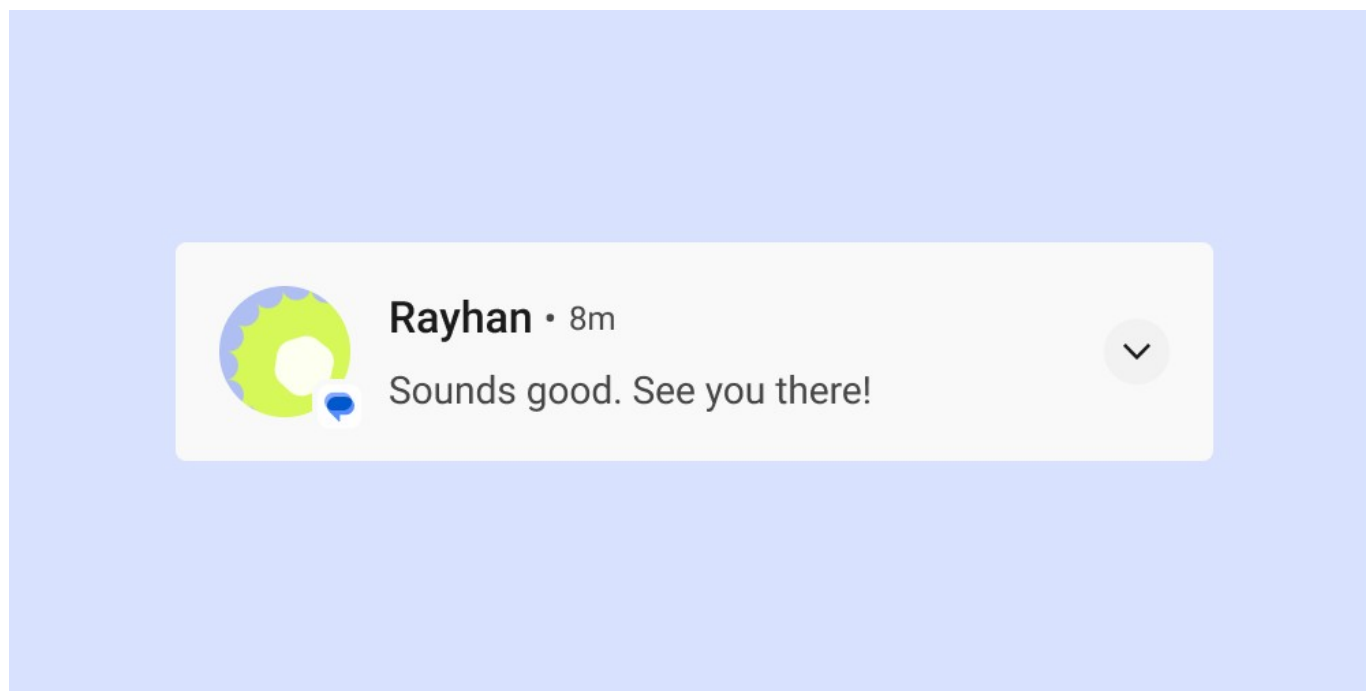
- Понять анатомию уведомления.
- Выбрать подходящий шаблон уведомления.
- Определить категорию уведомления, которая соответствует его назначению

Анатомия

Как устроены уведомления

Анатомия уведомления

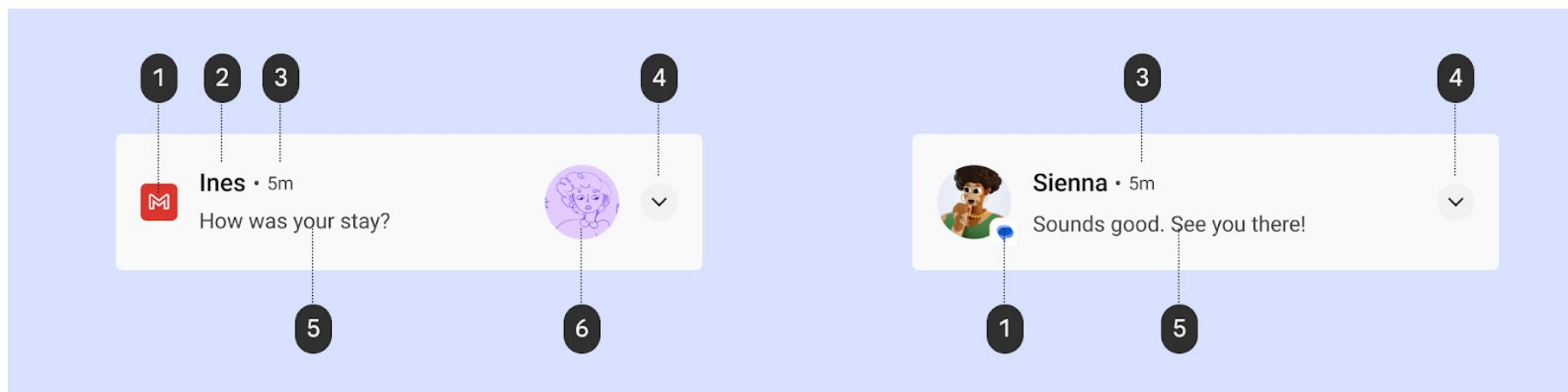
- Основной контент
- Люди
- Действия



Анатомия уведомления

1. Иконка приложения
2. Заголовок
3. Таймстемп (время)

4. Индикатор расширения
5. Текст содержимого
6. Большая иконка



Анатомия уведомления.

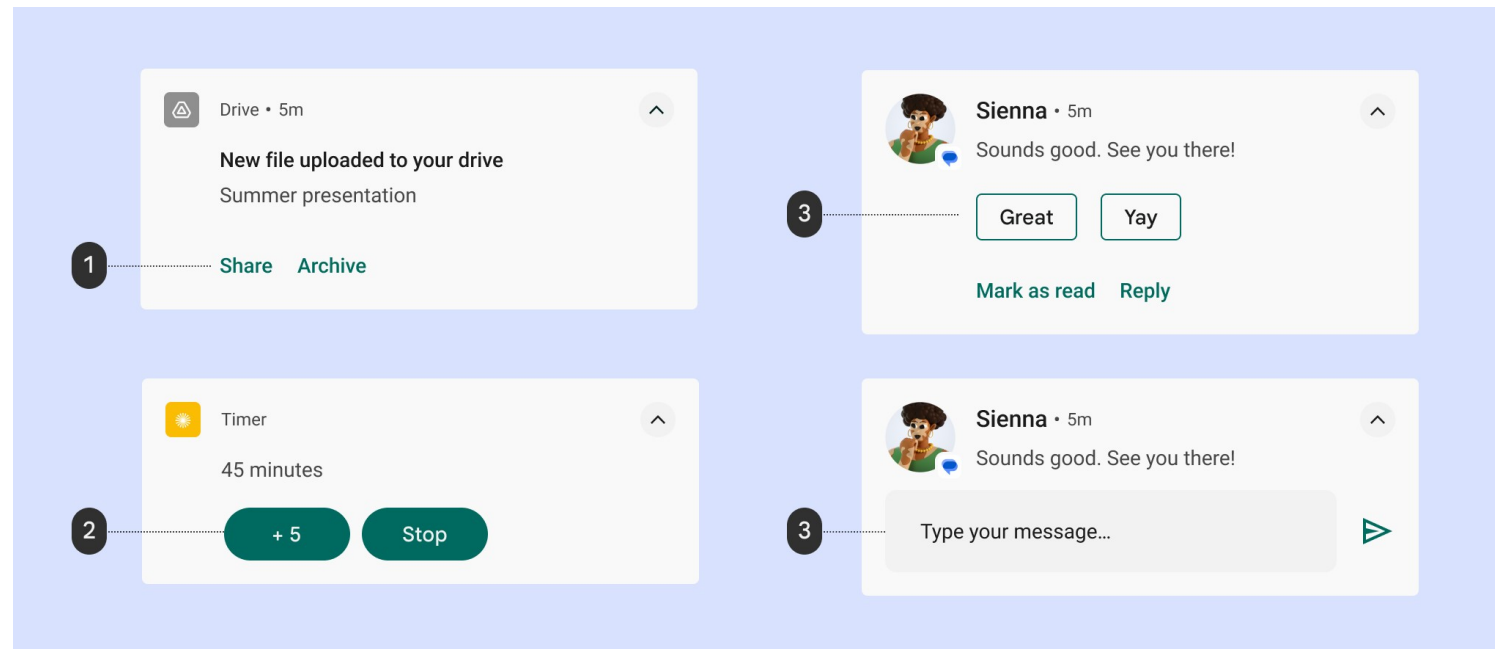
Действия с уведомлениями (Actions)

1. Кнопка с текстом

2. Заполненные кнопки действий

3. Предложенные ответы

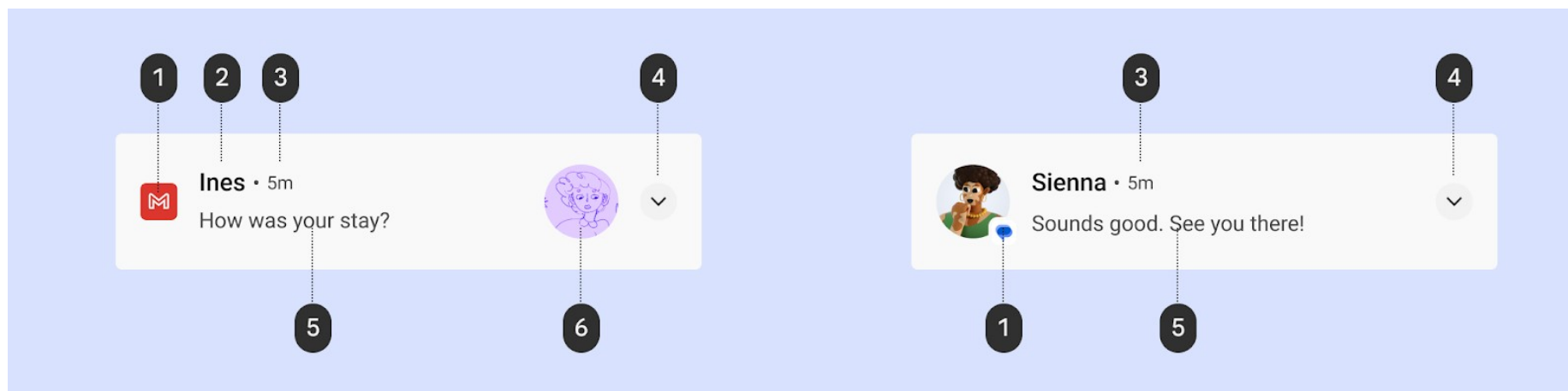
4. Reply



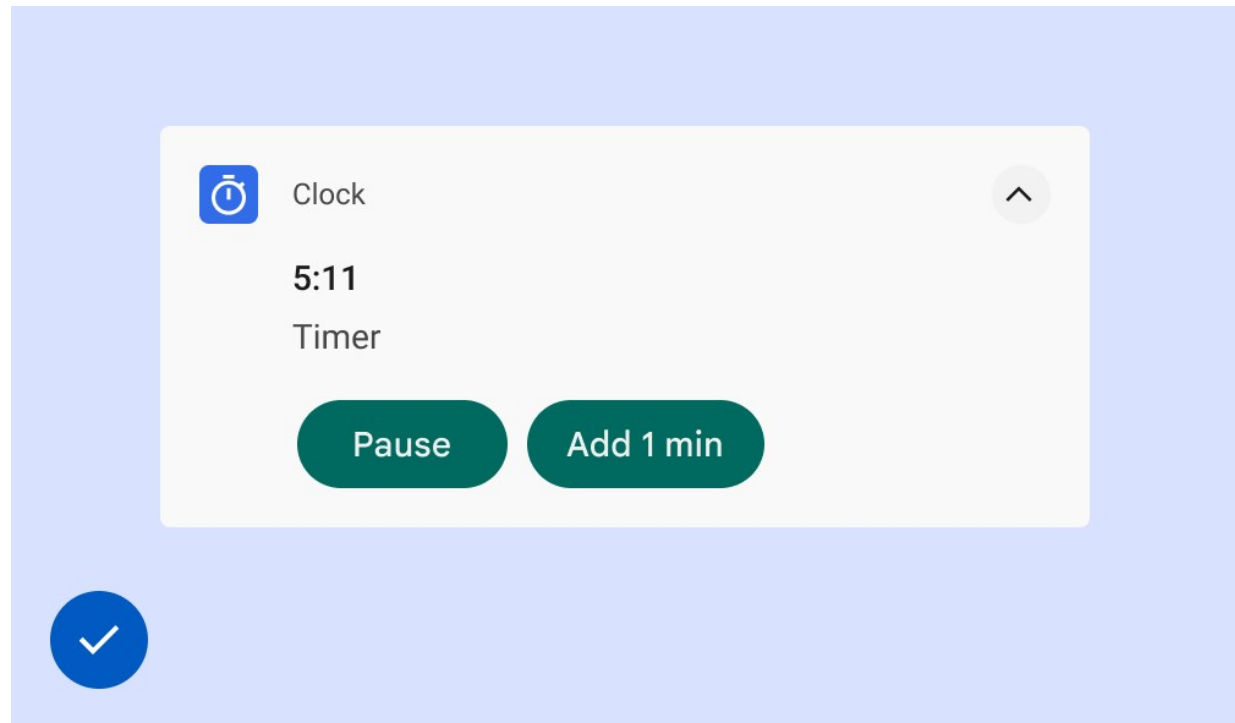
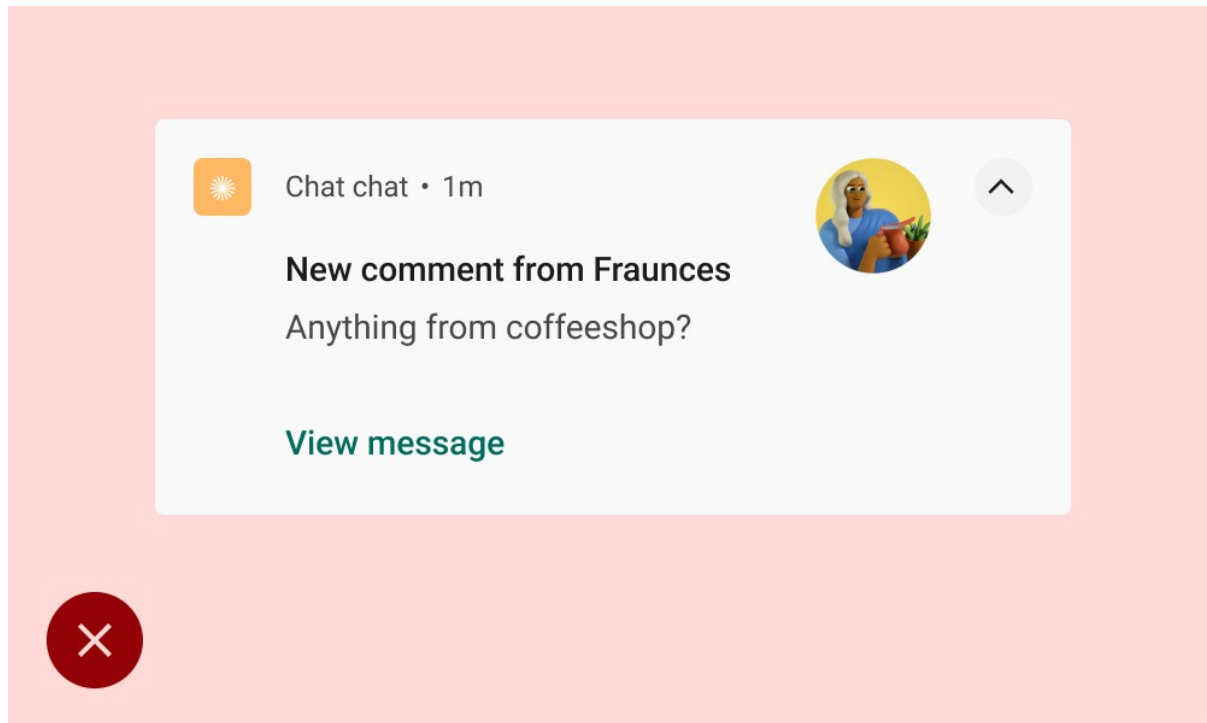
Анатомия уведомления

1. Иконка приложения
2. Заголовок
3. Таймстемп (время)

4. Индикатор расширения
5. Текст содержимого
6. Большая иконка



Анатомия уведомления. Расширенные представления (Expanded views)



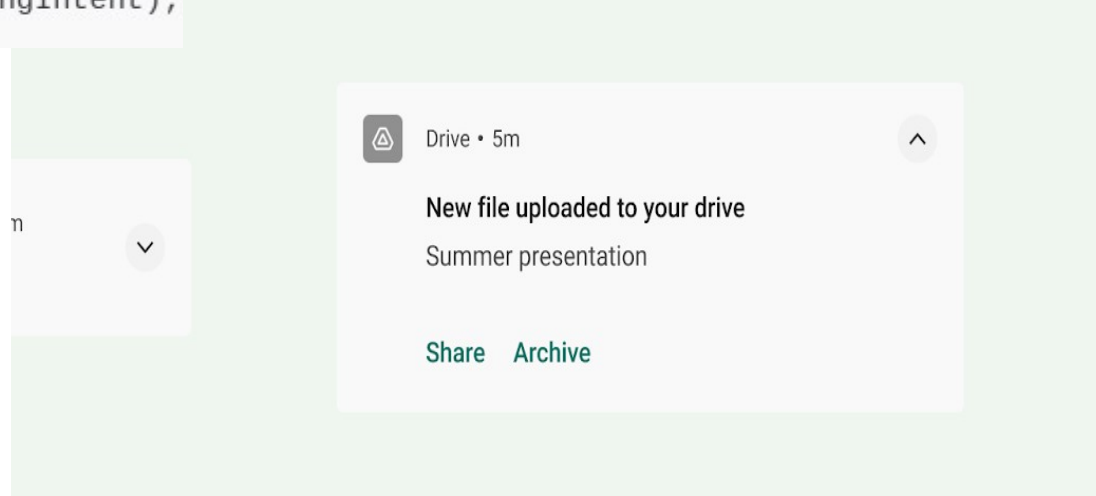
Шаблоны уведомлений

Какие шаблоны (стили) предоставляет API

Шаблоны уведомлений. Стандарт

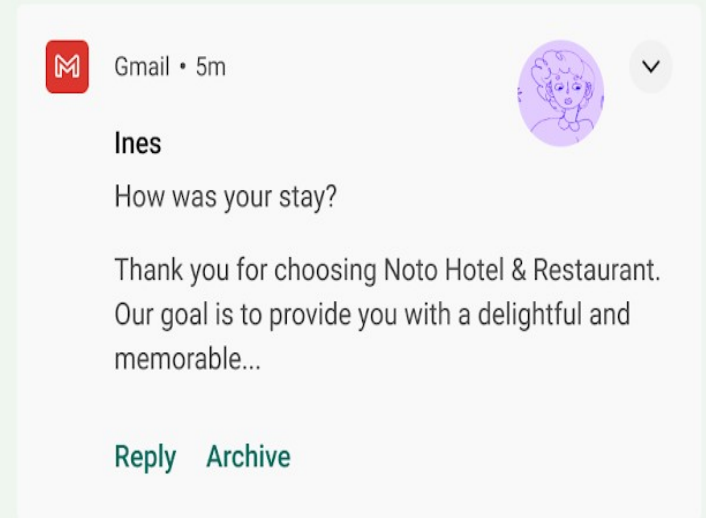
// Создаем уведомление

```
NotificationCompat.Builder builder =  
    new NotificationCompat.Builder(context, "channel_id")  
        .setSmallIcon(R.drawable.ic_notification)  
        .setContentTitle("New file uploaded to your drive")  
        .setContentText("Summer Presentation")  
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)  
    // Добавляем кнопки действий  
        .addAction(R.drawable.ic_share, "Share", sharePendingIntent)  
        .addAction(R.drawable.ic_archive, "Archive", archivePendingIntent);
```



Шаблоны уведомлений. Большой текст (Big text)

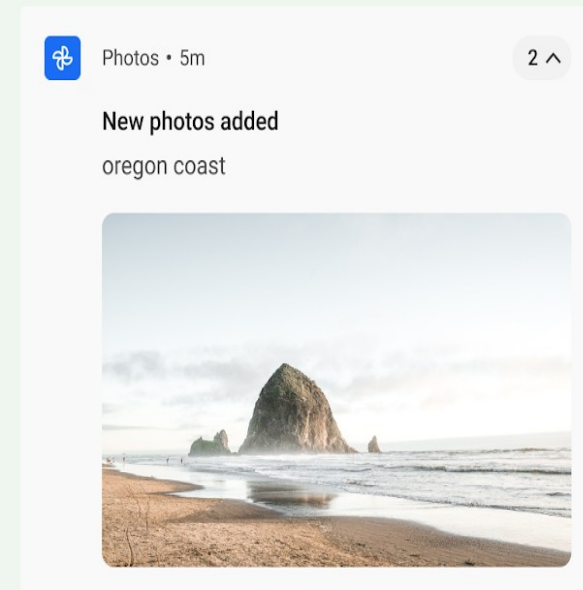
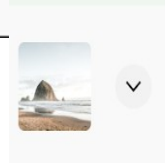
```
NotificationCompat.Builder builder =  
    new NotificationCompat.Builder(context, "channel_messages")  
        .setSmallIcon(R.drawable.ic_notification) // маленькая иконка (в статусе)  
        .setLargeIcon(senderAvatar) // аватар отправителя  
        .setContentTitle("Ines") // отправитель  
        .setContentText("How was your stay?") // краткий текст  
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)  
        // BigTextStyle для расширенного текста  
        .setStyle(new NotificationCompat.BigTextStyle()  
            .bigText("Thank you for choosing Noto Hotel & Restaurant. " +  
                "We hope you enjoyed your stay. " +  
                "Please take a moment to share your feedback."))  
        // Кнопки действий  
        .addAction(R.drawable.ic_reply, "Reply", replyPendingIntent)  
        .addAction(R.drawable.ic_archive, "Archive", archivePendingIntent);
```



Шаблоны уведомлений.

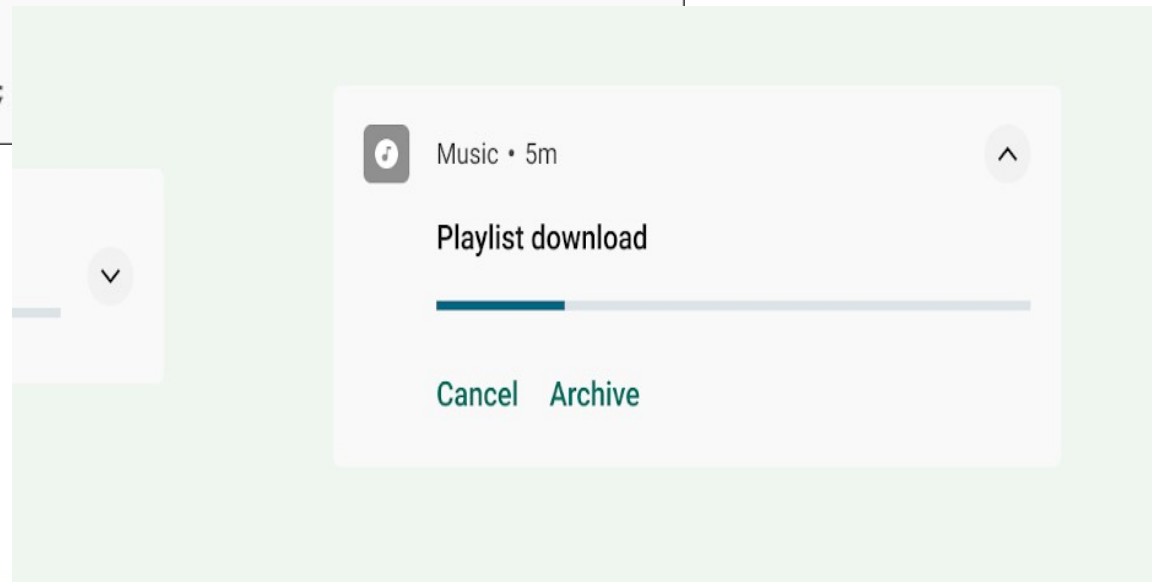
Большая картинка (Big picture)

```
NotificationCompat.Builder builder =  
    new NotificationCompat.Builder(context, "channel_messages")  
        .setSmallIcon(R.drawable.ic_notification) // маленькая иконка в статус-баре  
        .setContentTitle("New photos added")      // заголовок уведомления  
        .setContentText("Oregon coast")           // краткий текст  
        .setLargeIcon(largeIcon)                  // большая иконка в уведомлении  
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)  
        // BigPictureStyle с большой картинкой и показываем LargeIcon placeholder  
        .setStyle(new NotificationCompat.BigPictureStyle()  
            .bigPicture(bigPicture));
```



Шаблоны уведомлений. Прогресс (Progress)

```
NotificationCompat.Builder builder =  
    new NotificationCompat.Builder(context, "channel_music")  
        .setSmallIcon(R.drawable.ic_music_note) // маленькая иконка  
        .setContentTitle("Music") // название приложения  
        .setContentText("Playlist download") // контент  
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)  
        .setProgress(100, 20, false) // max=100, progress=20%, false = неиндетерминированный  
        // Кнопки действий  
        .addAction(R.drawable.ic_cancel, "Cancel", cancelIntent)  
        .addAction(R.drawable.ic_archive, "Archive", archiveIntent);
```



Отправка уведомлений

Как сказать ОС, что пора отправлять сформированные уведомления?

Отправка уведомлений

```
NotificationManagerCompat notificationManager = NotificationManagerCompat.from(context);  
notificationManager.notify(2, builder.build());
```

NotificationManagerCompat.from(context)


 Это синглтон для работы с уведомлениями.

 context - это обычно Activity, Service или Application. Через него Android знает, к какому приложению относится уведомление.

notificationManager.notify(int id, Notification n)

 Id - это уникальный идентификатор уведомления в рамках приложения.

Если уведомление с таким ID уже существует, оно будет обновлено, а не создано заново

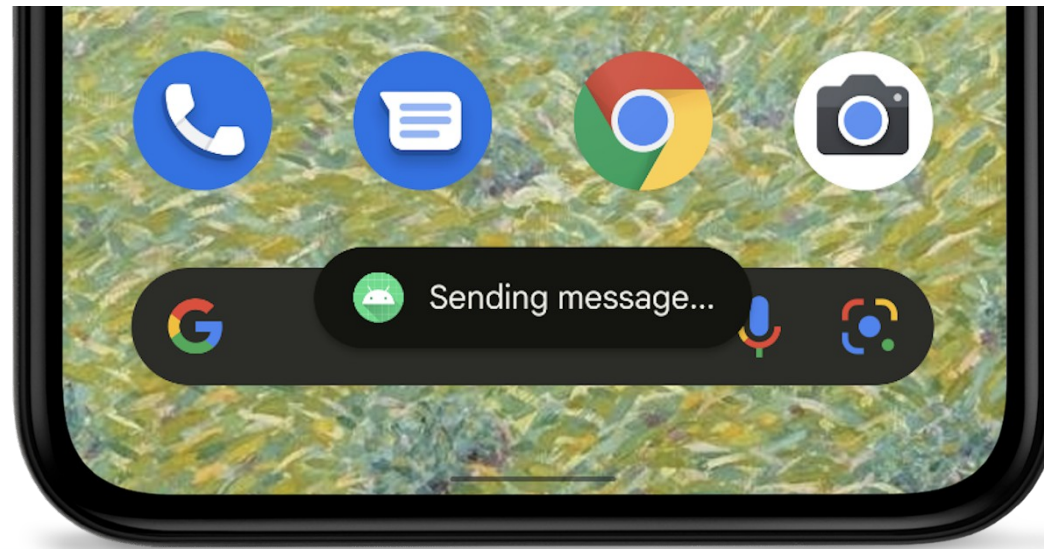
 build() собирает все параметры в полноценное уведомление, которое Android может отобразить

Тосты (Toasts)

Зачем они нужны и как их использовать

Тосты

Тост позволяет отобразить простой фидбек операции в маленьком рорир. Оно занимает немного экранного места, требуемого для сообщения, а текущая activity остается видимой и интерактивной.



Тосты. Создание

Чтобы инстанцировать тост, нужно использовать статический `makeText()` метод класса `Toast`. Он принимает следующие параметры: `activity` (контекст), текст сообщения и длительность сообщения.

```
CharSequence text = "Hello toast!";  
int duration = Toast.LENGTH_SHORT;  
  
Toast toast = Toast.makeText(this /* MyActivity */, text, duration);  
toast.show();
```

Вывод

**В ходе выполненной работы,
познакомились с понятиями
уведомления и toast, а также
научились их вызывать, используя
Android API**