Національний технічний університет України
“Київський політехнічний інститут ім І. Сікорського”

Звіт

з лабораторноі роботи №7
на тему “Посилання нотифікацій”

Виконав: студент групи
ІС-42 Коростильов Євгеній

Київ 2017 р

Нотифікації

Для точного означення нотифікацій звернемось до Вікіпедії:

Notifications - це повідомлення, які користувач бачить у верхній частині екрану, коли йому приходить новий лист, повідомлення, оновлення і т.п.

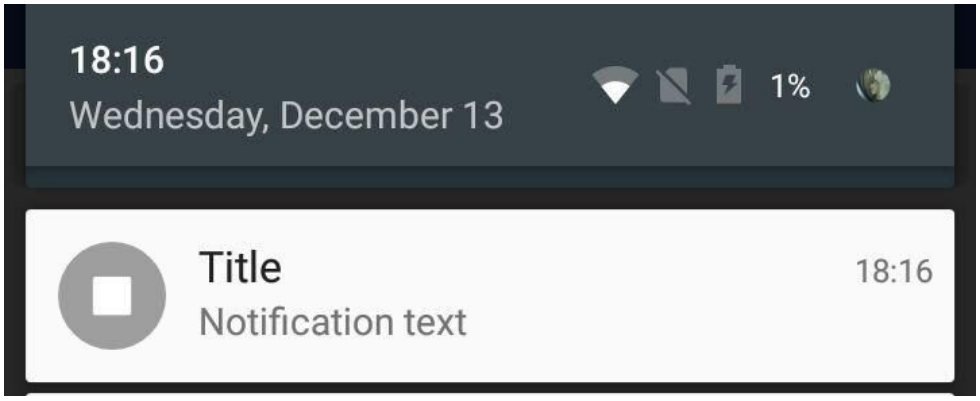
Тобто нотифікациі потрбіні для доставки повідомлень користувачу на екран. В випадку мого додатку я вирішив виводити в нотифікація рекорд в грі з пропозицією його перевершити.

Спочатку додав найпростішу структуру показу нотифікацій:

```
NotificationCompat.Builder builder =
       new NotificationCompat.Builder(this)
                .setSmallIcon(android.R.drawable.ic_dialog_email)
                .setContentTitle("Title")
                .setContentText("Notification text");

Notification notification = builder.build();

NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.notify(1, notification);
```



Розберемо цей приклад коду. Спочатку ми створюєму саму нотифікацюю. За допомогою SetSmallIcon додаємо іконку. SetContentTitle додає назву нотифікациі (тут краще всього вивоодити назву додатку), SetContentText задає саме зміст нотифікациі (тут будемо виводити наш рекорд).

За допомогою методу build() створюємо готову нотифікацію.

Далі використовуємо NotificationManager і його метод notify, щоб показати створене повідомлення. Крім notification, потрібно передати id. Це необхідно, щоб в подальшому ми могли використовувати цей id для поновлення або видалення повідомлення.

Оновлення

Можемо оновити цю нотифікацію за допомогою того ж самого id

```
NotificationCompat.Builder builder =
       new NotificationCompat.Builder(this)
               .setSmallIcon(android.R.drawable.ic_dialog_email)
               .setContentTitle("Title change")
               .setContentText("Notification text change");
 Notification notification = builder.build();

NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.notify(1, notification);
```

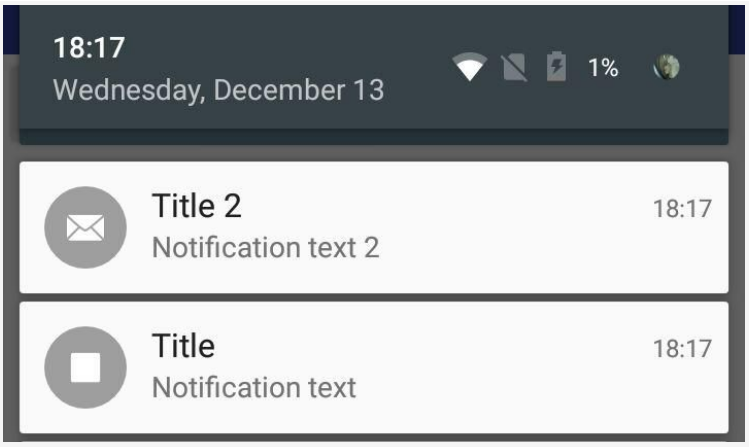
В цьому прикладі ми використовували такий ж самий код, за допомогою того, що ми не змінили id текст і заголовок оновився.

Декілька повідомлень

Щоб відправити декілька повідомлень треба декілька раз прописати нотифікації і дати ім різні id. В такому випадку прийде декілька різних повідомлень. Наведемо приклад:

```
NotificationCompat.Builder builder =
       new NotificationCompat.Builder(this)
               .setSmallIcon(R.mipmap.ic_launcher)
               .setContentTitle("Title")
               .setContentText("Notification text");
Notification notification = builder.build();
NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.notify(1, notification);


NotificationCompat.Builder builder =
       new NotificationCompat.Builder(this)
               .setSmallIcon(android.R.drawable.ic_dialog_email)
               .setContentTitle("Title 2")
               .setContentText("Notification text 2");
Notification notification = builder.build();
NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.notify(2, notification);
```



Видалення

Можна видалити повідомлення за id:

```
NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.cancel(1);
```

Також можливо видалити одразу всі повідомлення:

```
NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.cancelAll();
```

При видаленні повідомлення немає необхідності перевіряти, відображається воно чи ні. Якщо повідомлення з якихось причин вже немає, то просто нічого не відбудеться.

Обробка натискання

Щоб виконати будь-яку дію після натискання на повідомлення, необхідно використовувати PendingIntent. PendingIntent - це контейнер для Intent. Цей контейнер може бути використаний для подальшого запуску вкладеного в нього Intent.

Ми будемо створювати Intent для запуску, наприклад, Activity, упаковувати цей Intent в PendingIntent і передавати PeningIntent в повідомлення. При натисканні на повідомлення, система дістане з нього PedningIntent і використовує вкладений в нього Intent, щоб запустити Activity.

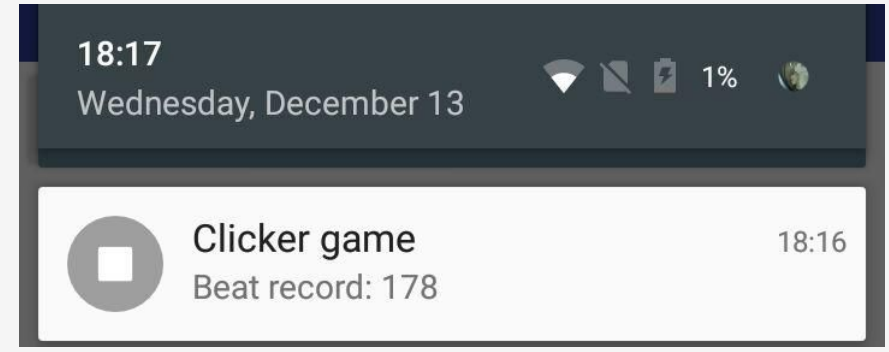
Приклад обробки нотифікації

```
Intent resultIntent = new Intent(this, MainActivity.class);
PendingIntent resultPendingIntent = PendingIntent.getActivity(this, 0,
resultIntent,PendingIntent.FLAG_UPDATE_CURRENT);
```

Його треба додати до тексту нотифікації.

```
Intent resultIntent = new Intent(this, MainActivity.class);
PendingIntent resultPendingIntent = PendingIntent.getActivity(this, 0,
resultIntent,PendingIntent.FLAG_UPDATE_CURRENT);

NotificationCompat.Builder builder = new
NotificationCompat.Builder(this)
                .setSmallIcon(R.mipmap.ic_launcher)
                .setContentTitle("Title")
                .setContentText("Notification text")
                .setContentIntent(resultPendingIntent)
                 .setAutoCancel(true);
```




```
Notification notification = builder.build();

NotificationManager notificationManager =
       (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notificationManager.notify(1, notification);
```

Після цього буде оброблюватись нотифікація і відкриється головне активіті.

Висновки

Як видно зі звіту Android API надає дуже широкий спектр технологій для роботи з нотифікаціями. Можливо надсилати, оновлювати, видаляти та кастомізувати вигляд нотифікації - як іконок так і стилю.