



通知チャンネル機能と バッチ表示について

- 武藤繁夫
 - フリーランスプログラマ (TM 研究所)
 - 暇です。仕事待ってます。
 - JAG コミュニティ運営委員、同 ML モデレータ、福祉部
- 本日の発表について
 - Notification Channel, dots と呼びますよ
 - それは何か、全体像の把握
 - コード触ってみたり
 - 気をつけるべき点など

Notification に関する歴史

OS Version	Topics
4.4 Kitkat 以前	制御一切なし。
5.x Lollipop	設定:「音と通知」から'すべてブロック'が可能に
6.x Marshmallow	'瞬間表示を許可'とか増えた
7.x Nougat	'マナー中','ロック画面'で許可とか増えた
8.0 Oreo	設定:「アプリと通知」からカテゴリ (Channel) ごと詳細に設定できるようになった

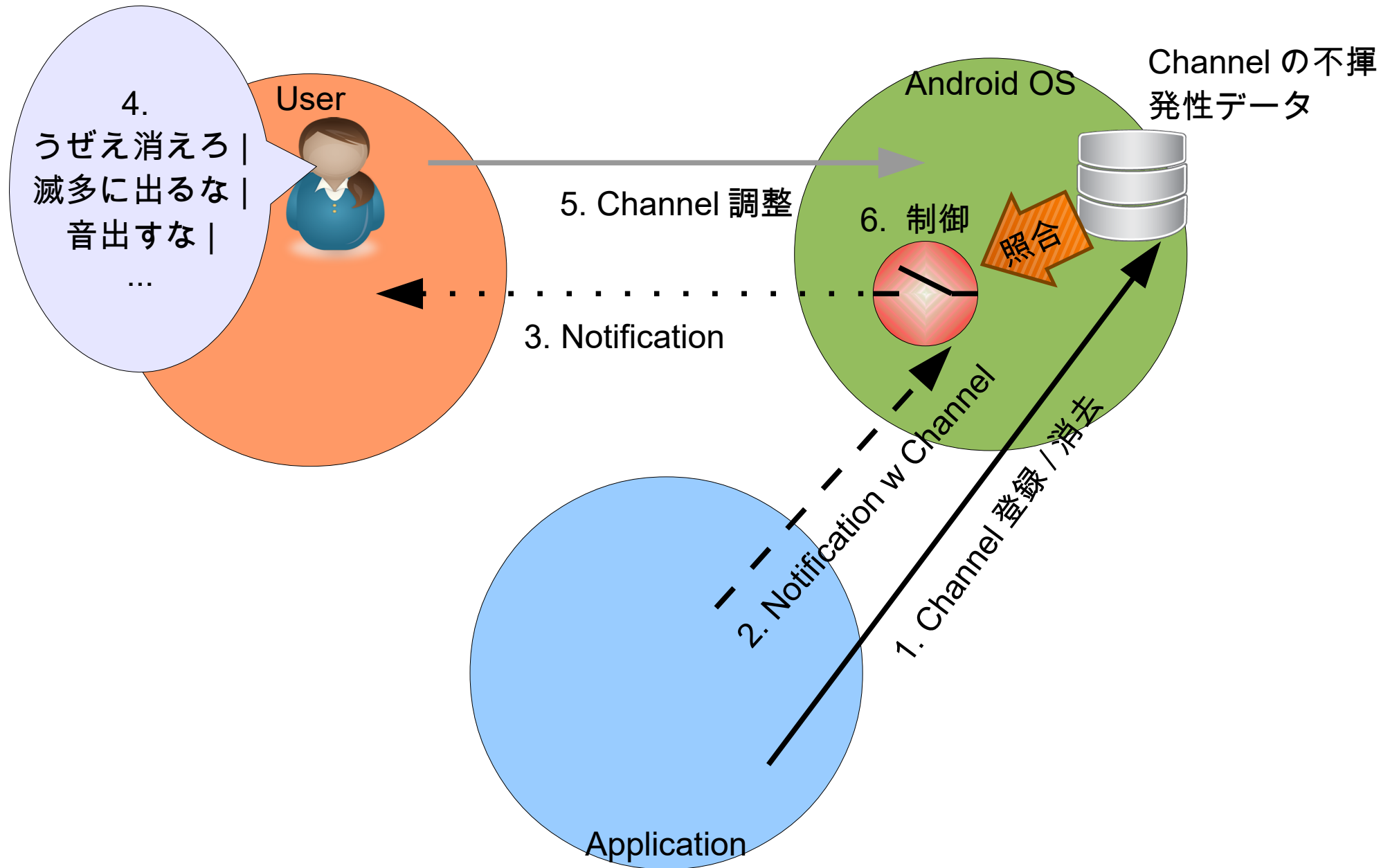
Notification Channel の導入。
ずっとドタバタしてたが、
このへんがいい落とし所でしょう。

Notification Channel とは？

- Notification のラベリングと、ユーザー主体の通知調整機構の総称

サンプルプログラムを動かしてみましょう。

Notification Channel 全体像



プログラマーがすること

- NotificationChannel を生成、登録
- Notification.Builder に ChannelID をセットする

サンプルコードを追ってみましょう。

コードスニペット

```
//use sdk26
import android.app.Notification;
import android.app.NotificationChannel;
...

manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
if( oreo ) {
    // Notification Channelを登録する
    NotificationChannel chan1 = new NotificationChannel(PRIMARY_CHANNEL,
        getString(R.string.noti_channel_default), NotificationManager.IMPORTANCE_DEFAULT);
    chan1.setLightColor(Color.GREEN);
    chan1.setLockscreenVisibility(Notification.VISIBILITY_PRIVATE);
    manager.createNotificationChannel(chan1);
}

{
    Notification.Builder b = new Notification.Builder(getApplicationContext())
        .setContentTitle("mytitle")
        .setContentText("mytext")
        .setSmallIcon(getSmallIcon())
        .setAutoCancel(true);
    if( oreo ) {
        b.setChannelId(PRIMARY_CHANNEL);
    }
    manager.notify(id, b.build());
}
```

だよねだよね、
横線引かれるよね

はあ？ 後方互換どうすんだよ NotificationCompat 使えよ

- ざんねーん、supportlib26.0.0-alpha1 は未対応！
- じゃあこうするしかないのか

```
private Notification compatTest(String title, String body) {  
    Notification n = null;  
    if( Build.VERSION.SDK_INT >= Build.VERSION_CODES.O ) {  
        n = new Notification.Builder(getApplicationContext(), PRIMARY_CHANNEL)  
            .setContentTitle(title)  
            .setContentText(body)  
            .setSmallIcon(getSmallIcon())  
            .setAutoCancel(true)  
            .build();  
    } else {  
        n = new NotificationCompat.Builder(getApplicationContext())  
            .setContentTitle(title)  
            .setContentText(body)  
            .setSmallIcon(getSmallIcon())  
            .setAutoCancel(true)  
            .build();  
    }  
    return(n);  
}
```

一瞬かんたんに見えるけど ...

- 長期的には、データ大丈夫なのかって普通に疑問がわくよね
- いくつか実験を試してみた

どっちが勝つか問題

- ユーザーが設定変更した後、createChannelしたらアプリとユーザーのどっちが勝つ？

ユーザー

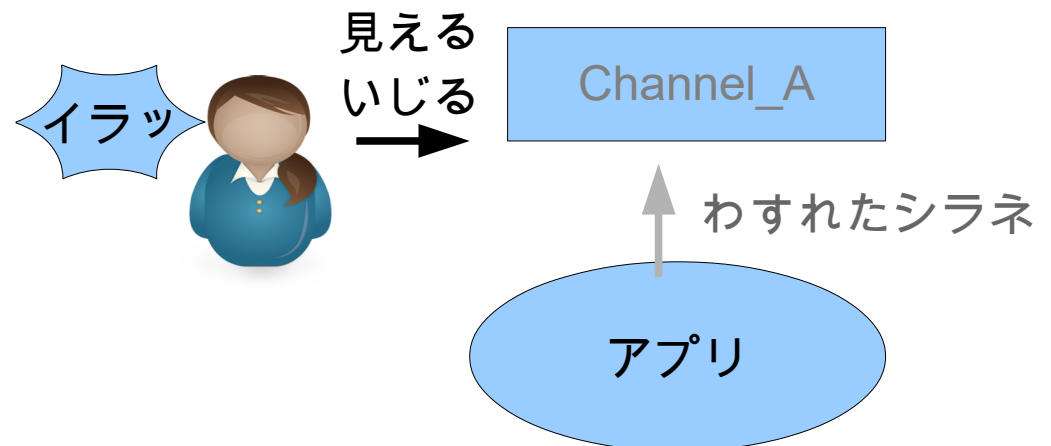
- 一度 deleteChannelした Channel を復活させたとき、アプリとユーザー（の過去の設定値）のどっちが勝つ？

ユーザー

つまり Channel ID は永続性がある

みなしご Channel ID 問題

- みなしご発生シーケンス
 - アプリが Chanell_A を生成
 - ある日アプリのソースコードから Channel_A が消滅した version を release して Channel_A を完全に忘却
 - ユーザーから見れば、Channel 設定には Channel_A は残っているが、もはや意味がない上に誰も消去できない。uninstall されるまで。



そもそも使わないという選択は？

- *target sdk が Android O 以降は Notification Channel を使わないと Notification ができません！*
 - 何もいわずにひっそりと表示されない
 - 将来、テスト漏れの可能性大
 - だから将来的には Notification Channel の実装は避けて通れない。Notification を出したければ。
- 今日一大事です
 - 忘れないでね。僕は忘れると思います！

使用上の注意

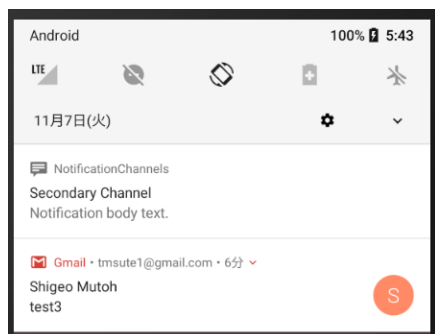
- アプリは Channel ID を未来永劫、継続的に管理しないといけない
 - みなしご Channel を発生させたくないから
 - Channel ID をハードコードすると死ねますよ
 - 安易には、アプリ起動時に存在する全 Channel の消去を行い、必要な Channel を新規に生成すれば解決するだろう。Channel ID データは永続性があるから大丈夫だ。
 - やって見たらうまくいった。お勧めしないけど、そうでもないときっとみなしごは発生するでしょう。

＜おまけ＞ クズアプリの例

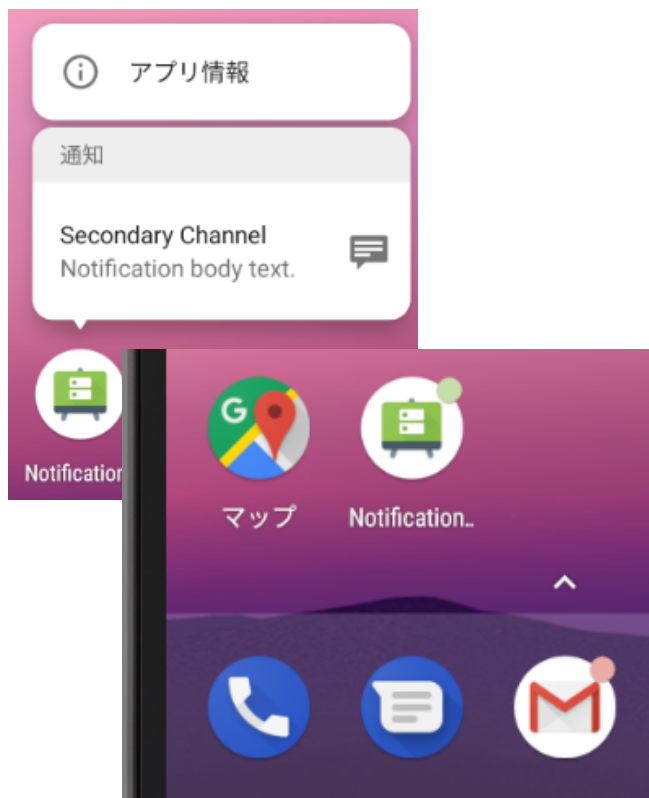
- Notification を投げる度に新しい id の NotificationChannel を作ってしまえば、ユーザー無視のクズアプリになれる気がする。。
 - まじでやめて
 - まあ一括 OFF されるか、uninstall されるだけ

次は Notification dots

Notification dots(badges) について



- アイコン右上のひとだまみたいなやつ
- Notification と連動して発生
- ひとだまの色はアイコンの色から自動生成
- ユーザーは Channel 設定から出さないようにもできる
- アイコン長押しで吹き出しが出る



前からあったよね

1

- 確かにあった。未読数が表示されてたり。
- しかしカラクリは、各 launcher ごとに定義された intent を、アプリが相手に合わせて broadcast するという、にわかに信じ難い実装により実現
 - ある launcher は key=unreadnum を、ある launcher は key=unreadcount を、てな感じ。
- だから主要 launcher に対応した library があって、それを使うと必要な intent を投げてくれたりする

で、Oreo の Notification dots は 素晴らしいのか？

- launcher が icon overlay を実装するのが楽になった
- アプリが未読数 intent を投げなくてもよくなった
- けど tradeoff で未読数ではなく、ひとだま表示
- 3rd party launcher と比較した場合、未読数にどこまでこだわるかで決まる

Oreo 以降の Notification Channel および dots に関するまとめ

- Notification のラベリング機構として channel を発明
- Icon overlay に関する実装として dots を発明
- 両者を Notification 制御データとして OS レベルでサポートし、ユーザによる細かい制御が可能に

アプリ開発者の憂鬱度

- Notification Channel
 - いずれ無視できなくなる日が来るし、Channel ID の管理もあるので、Notification 全体を包括するクラスを書かなきゃなあ、と思う。
- dots
 - launcher を書く人以外には関係ないと言える。
 - 未読数の intent を投げてたアプリは一瞬悩むが結局投げ続けるんだと思う。

おしまい

ありがとうございました。