



មជ្ឈមណ្ឌលកូរ៉េ សហវៃរ អេច អ ឌី Korea Software HRD Center

ប្រធានបទ៖ **Local Notification**

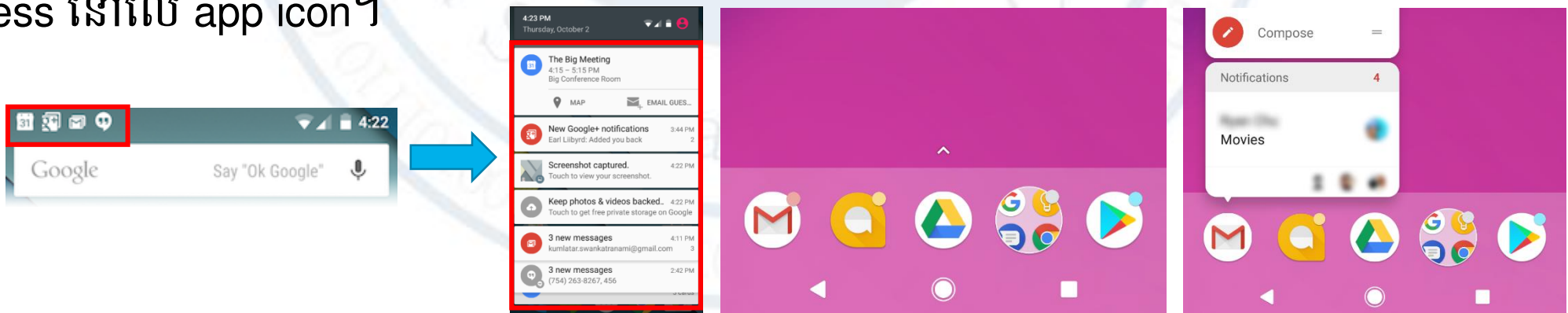
អ្នកប្រឹក្សាយោបល់: បណ្ឌិត គឹម ថេឡុង

មាតិកា

1. Status Bar Notification
2. Create Notification channel
3. Pending Intent

1. ស្វែងយល់ពី Status Bar Notification

- Notification គឺជា Message ដែលបង្ហាញទៅអ្នកប្រើប្រាស់ នៅខាងក្រៅនៃ application UI ដើម្បីបញ្ជាក់ថា មាន event អ្វីមួយបានកើតឡើង។
- នៅពេលយើងអោយ system ធ្វើនូវ Notification នោះវានឹងបង្ហាញជាលក្ខណៈ Icon នៅក្នុង Notification Area។ ដើម្បីឃើញនូវ detail notification យើងត្រូវបើកនូវ notification drawer។
- Android 8.0 (API level 26) គឺបានបន្ថែមនូវ feature ថ្មីគឺ យើងអាច alert notification នៅលើ Launcher icon របស់ App ដែលគេហៅថា notification badges។ ហើយយើងអាចមើល notification នោះបាន ដោយ long-press នៅលើ app icon។



1. ស្វែងយល់ពី Status Bar Notification

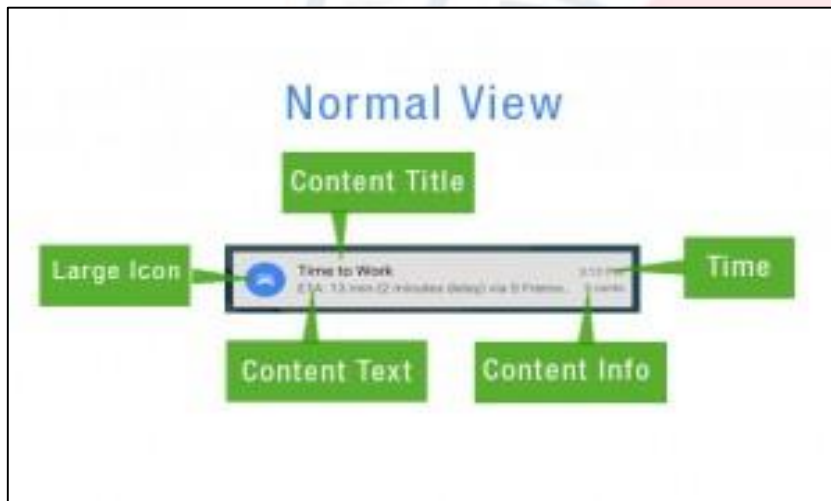
- យើងអាច specify នូវ UI Information និង Action សំរាប់ Notification នៅក្នុង NotificationCompat.Builder Object។
- ដើម្បីបង្កើត Notification យើងត្រូវ Call NotificationCompat.Builder.build() ដែលវា Return នូវ Notification Object។
- បន្ទាប់មកយើង Pass Notification Object ទៅកាន់ System ដោយការ Call NotificationManager.notify()

1. ស្វែងយល់ពី Status Bar Notification

- Notification object ត្រូវមាន៖
 - **setSmallIcon(int icon)** : to set a small icon
 - **setTitle(charSquencetext)** : to set a title
 - **setContentText(charSquencetext):** to set detail text
 - **setTicker(charSquencetext):** to set the text ដើម្បីបង្ហាញនៅពេល notification bar
 - **setLargeIcon(Bitmap icon)** : to set the large icon
 - **setDefault(int default)** : to set the default notification option
 - **setContentIntent(PendingIntent intent)** : to set PendingIntent នៅពេល user touch លើ Notification
 - **setContentInfo()** :to set text នៅខាងស្តាំនៃ notification

1. ស្វែងយល់ពី Status Bar Notification

រូបខាងក្រោមបញ្ជាក់ពីទម្រង់នៃ Normal view និង Big view នៃ Notification



1. ស្វែងយល់ពី Status Bar Notification

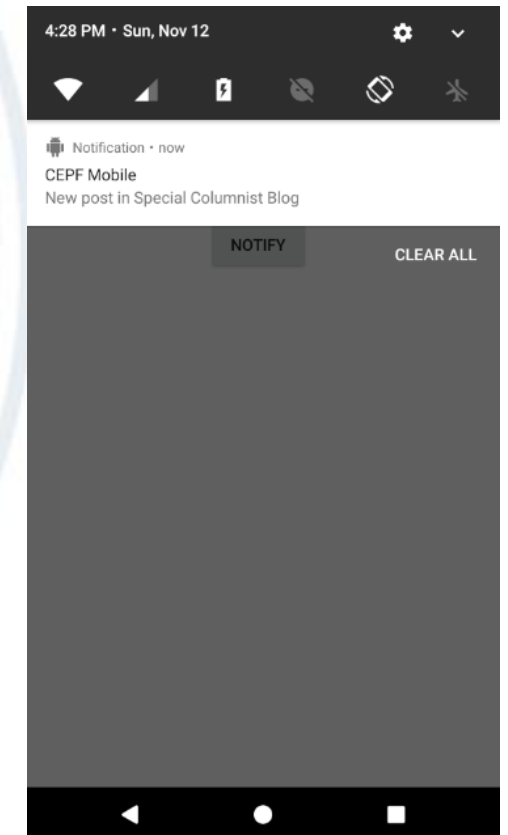
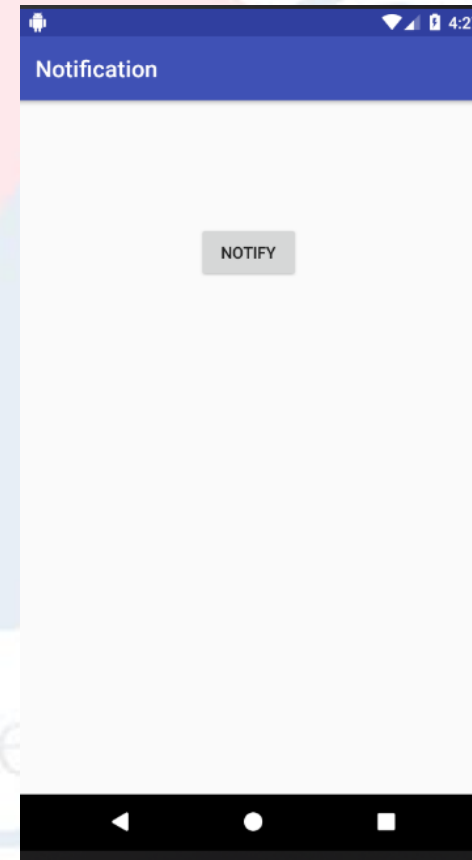
- Status Bar Notification បញ្ជាក់អំពីអ្វីដែលបានដំណើរការជាលក្ខណៈ background service ដូចជា Downloading ឬ Playing music ឬ display an alert/Information។
- Notifications ត្រូវបាន Handle ដោយ NotificationManager នៅក្នុង Android។ NotificationManger គឺជា System Service មួយដែលត្រូវបានប្រើក្នុងការគ្រប់គ្រង Notifications.
- ដើម្បីបង្កើត Status bar Notification យើងត្រូវការ Class ពីរគឺ **NotificationManager** និង **Notification** ។

1. វិធីសាស្ត្រ Status Bar Notification

ឧទាហរណ៍

```
button=findViewById(R.id.btnnotification);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Context context=MainActivity.this;
        Intent intent=new Intent(context,Result.class);
        NotificationCompat.Builder builder=new NotificationCompat.Builder(context);
        builder
            .setContentTitle("CEPF Mobile")
            .setLargeIcon(BitmapFactory.decodeResource
                (getResources(),R.drawable.ic_audio))
            .setSmallIcon(R.drawable.ic_android)
            .setContentText("New post in Special Columnist Blog")
            .setTicker("Ticker");
        NotificationManager nm= (NotificationManager)
            getSystemService(Context.NOTIFICATION_SERVICE);
        int notificationId= (int) (System.currentTimeMillis()/1000);
        nm.notify(notificationId,builder.build());
    }
});
```

When user click button it will make notification



2. Notification Channel

- សម្រាប់ Android 8.0 (API level 26) ឡើង គឺតម្រូវឱ្យបង្កើត Notification channel ដើម្បីធ្វើការ customize channel សម្រាប់ប្រភេទ notification នីមួយៗ ដែលចង់បង្ហាញទៅកាន់ User
- វាផ្តល់នូវ feature ដើម្បី user អាចធ្វើការ manage notifications ឧទាហរណ៍ ដូចជា User អាចធ្វើការ enable or disable ប្រភេទ notifications ដែលត្រូវបានបង្កើតដោយ Notification manager នៅក្នុង app របស់យើង
- គ្រប់ notifications ទាំងអស់ដែលបាន posted ចូលទៅ notification channel តែមួយ គឺវាមាន behavior តែមួយ
- យើងអាចធ្វើការកំណត់នូវ behavior ទៅ ទៅឱ្យ Notification Channelដូចជា៖
 - Impotence
 - Sound
 - Light
 - Show on lock screen
 - Override do not disturb

2. Notification Channel (ត)

- Create a notification channel
 - បង្កើត notification channel object ហើយកំណត់នៅ Unique ID ទៅឲ្យវា
 - Configure notification channel object ដូចជា alert sound, description ឬ Vibration ...
 - Submit the notification channel object ទៅកាន់ notification manager

```
@TargetApi(Build.VERSION_CODES.O)
public static <T extends Channel> void createNotificationChannel(Context context, T channel, String ... groupIds){
    android.app.NotificationChannel mChannel = new android.app.NotificationChannel(channel.getId(),channel.getName(),channel.getPriority());
    mChannel.setDescription(channel.getDesc());
    mChannel.enableLights(true);
    mChannel.setLightColor(Color.RED);
    mChannel.enableVibration(true);
    mChannel.setVibrationPattern(new long[]{100l, 200l, 300l, 400l, 500l, 400l, 300l, 200l, 400l});

    NotificationManager mNotificationManager= (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
    mNotificationManager.createNotificationChannel(mChannel);
}
```

2. Notification Channel (ត)

- Create a notification channel group
 - ប្រសិន បើ app support multiple user accounts យើងក៏អាចបង្កើត notification channel group មួយ ដើម្បីគ្រប់គ្រងលើ accounts នីមួយៗ។
 - វាផ្តល់ឱ្យ user អាចធ្វើការគ្រប់គ្រង multiple notification channels ដោយប្រើឈ្មោះជាក់លាក់ នៅលើ app តែមួយ។

```
@RequiresApi(api = Build.VERSION_CODES.O)
public static <T extends ChannelGroup> String createNotificationChannelGroup(Context context, T group){
    NotificationChannelGroup mGroup=new NotificationChannelGroup(group.getId(),group.getName());
    NotificationManager mNotificationManager= (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
    mNotificationManager.createNotificationChannelGroup(mGroup);
    return group.getId();
}
```

Then set group to each channel

```
if(groupIds!=null){
    mChannel.setGroup(groupIds[0]);
}
```

2. Notification Channel (ត)

- Delete a notification channel
 - ដើម្បី delete notification channel ដោយប្រើ method deleteNotificationChannel()

```
@RequiresApi(api = Build.VERSION_CODES.O)
public static void deleteChannel(Context context, String channelId){
    NotificationManager mNotificationManager= (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
    mNotificationManager.deleteNotificationChannel(channelId);
}
```

2. ស្វែងយល់អំពី Pending Intent

- PendingIntent គឺជា Class ដែលអាចធ្វើការបង្កើត Object ដើម្បីធ្វើការផ្ទុកនូវ Intent ផ្សេងទៀតដើម្បីអោយ Application ដទៃទៀតហៅវាយកទៅប្រើប្រាស់នៅពេលមាន Action កើតឡើង។
- ការផ្តល់នូវ PendingIntent ទៅអោយ Application ដទៃមានន័យថា Application របស់យើងផ្តល់សិទ្ធិ ដែលយើងមានទៅអោយ Application ដទៃនោះដើម្បីធ្វើការបើកនូវ Object Intent ដែលយើងបានកំណត់ Application។

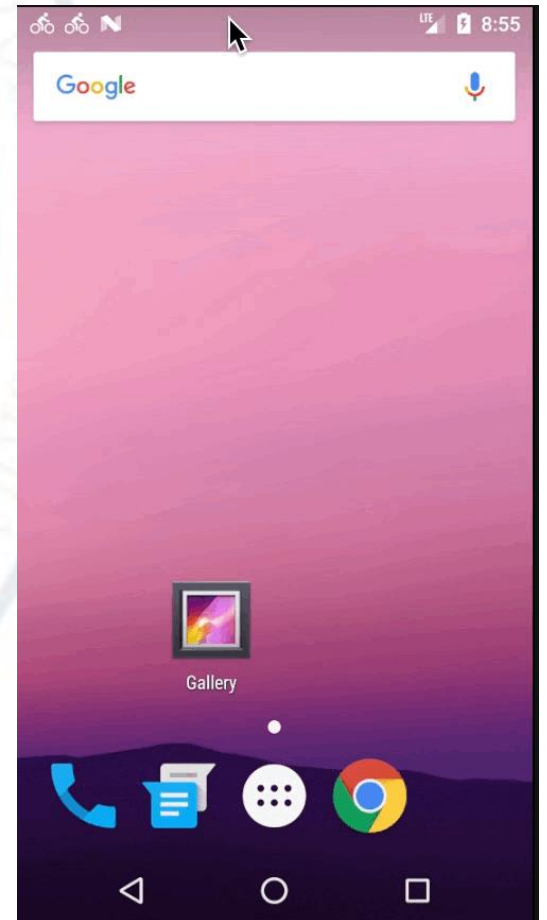
2. ស្វែងយល់អំពី Pending Intent

- ឧទាហរណ៍ App របស់យើងអនុញ្ញាតិអោយ NotificationManager ធ្វើការបើកbrowser ចូលទៅកាន់

<https://facebook.com>

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://facebook.com"));
PendingIntent pendingIntent = PendingIntent.getActivity(context: MainActivity.this, requestCode: 1,
                                                    intent, PendingIntent.FLAG_UPDATE_CURRENT);
Uri defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(),
                                                    channelId: "my_channel_01");
builder
    .setSmallIcon(R.drawable.ic_bike)
    .setAutoCancel(true)
    .setContentTitle("New notification from Facebook")
    .setContentText("Someone send you a friend request")
    .setContentInfo("Hello this is content info")
    .setSound(defaultSoundUri)
    .setPriority(Notification.PRIORITY_HIGH)
    .setDefaults(Notification.DEFAULT_VIBRATE)
    .setSubText("Hello this is a sub text")
    .setContentIntent(pendingIntent);

NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
int notificationId = (int) (System.currentTimeMillis()%1000);
notificationManager.notify(notificationId, builder.build());
```



ಸ್ವಪನ್ನಜ್ಞಾನ!!!