

Tecnologías y Desarrollo en Dispositivos Móviles

Apartado 16: Notificaciones

Autor:
Miguel Ángel García Cumbreiras

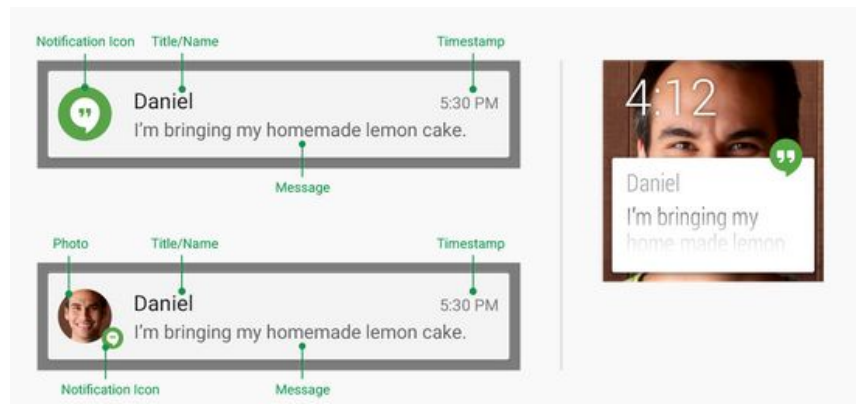
Introducción

- Las Notificaciones Push son aquellos mensajes que recibimos en el dispositivo y que han sido emitidos desde cualquier punto de un sistema.
- Actores:
 - Sistema de mensajería: El servicio web para el envío de Notificaciones Push a dispositivos Android.
 - Servidor: con un servicio (REST, SOAP, aplicación web, etc...) que será el encargado de gestionar los identificadores de registro de dispositivos a los que podemos enviar las notificaciones y de comunicarse con el sistema de mensajería solicitando el envío de notificaciones al dispositivo (o dispositivos) deseados.
 - Dispositivo Android: que recibirá las notificaciones.

Componentes de una notificación

- Icono de notificación, normalmente el de la propia app (setSmallIcon())
- Título de la notificación (setContentTitle())
- Texto (setContentText())
- Timestamp

Nota: Material Design (Android 5.0, API 21) introduce novedades de diseño en la guía de diseño de apps Android.



Componentes de una notificación

- Opcionalmente podemos asociar acciones a una notificación, si no es la acción normal de abrir la app, y prioridad.

<https://elandroidelibre.lespanol.com/2017/03/android-o-distinguir-entre-tipos-de-notificacion-para-una-mejor-gestion.html>

<https://www.tuexperto.com/2017/04/06/asi-sera-el-nuevo-sistema-de-notificaciones-en-android/>

Prioridades de una notificación

- Android permite indicar una prioridad:

Priority	Use
MAX	Use for critical and urgent notifications that alert the user to a condition that is time-critical or needs to be resolved before they can continue with a particular task.
HIGH	Use primarily for important communication, such as message or chat events with content that is particularly interesting for the user. High-priority notifications trigger the heads-up notification display.
DEFAULT	Use for all notifications that don't fall into any of the other priorities described here.
LOW	Use for notifications that you want the user to be informed about, but that are less urgent. Low-priority notifications tend to show up at the bottom of the list, which makes them a good choice for things like public or undirected social updates: The user has asked to be notified about them, but these notifications should never take precedence over urgent or direct communication.
MIN	Use for contextual or background information such as weather information or contextual location information. Minimum-priority notifications do not appear in the status bar. The user discovers them on expanding the notification shade.

MAX	HIGH	DEFAULT	LOW	MIN
very urgent and time-critical	most notifications go here			opportunistic
incoming/ongoing calls turn-by-turn directions emergency alert	Important emails realtime chat SMS	everything else (and all legacy notifications)	software updates	expired events suggestions weather location info

Categorías de una notificación

CATEGORY_CALL	Incoming call (voice or video) or similar synchronous communication request
CATEGORY_MESSAGE	Incoming direct message (SMS, instant message, etc.)
CATEGORY_EMAIL	Asynchronous bulk message (email)
CATEGORY_EVENT	Calendar event
CATEGORY_PROMO	Promotion or advertisement
CATEGORY_ALARM	Alarm or timer
CATEGORY_PROGRESS	Progress of a long-running background operation
CATEGORY_SOCIAL	Social network or sharing update
CATEGORY_ERROR	Error in background operation or authentication status
CATEGORY_TRANSPORT	Media transport control for playback
CATEGORY_SYSTEM	System or device status update. Reserved for system use.
CATEGORY_SERVICE	Indication of running background service
CATEGORY_RECOMMENDATION	A specific, timely recommendation for a single thing. For example, a news app might want to recommend a news story it believes the user will want to read next.
CATEGORY_STATUS	Ongoing information about device or contextual status

Clases principales

Toda la parte UI se incluye en un objeto *Notification* mediante la llamada a *NotificationCompat.Builder*, y la función `build()` nos devolverá un objeto *Notification* que contiene toda esa especificación.

Lo mínimo a incluir en *Notification* es:

- Un icono, mediante `setSmallIcon()`
- Un título, mediante `setContentTitle()`
- Un texto detallado, mediante `setContentText()`

Veamos un ejemplo de código...

Ejemplo

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle(textTitle)
    .setContentText(textContent)
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle("My notification")
    .setContentText("Much longer text that cannot fit one line...")
    .setStyle(new NotificationCompat.BigTextStyle()
        .bigText("Much longer text that cannot fit one line..."))
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

....

```
NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
// notificationId is a unique int for each notification that you must define
notificationManager.notify(notificationId, mBuilder.build());
```


Ejemplo (channel)

```
private void createNotificationChannel() {  
    // Create the NotificationChannel, but only on API 26+ because  
    // the NotificationChannel class is new and not in the support library  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        CharSequence name = getString(R.string.channel_name);  
        String description = getString(R.string.channel_description);  
        int importance = NotificationManager.IMPORTANCE_DEFAULT;  
        NotificationChannel channel = new NotificationChannel(CHANNEL_ID,  
name, importance);  
        channel.setDescription(description);  
        // Register the channel with the system; you can't change the  
importance  
        // or other notification behaviors after this  
        NotificationManager notificationManager =  
getSystemService(NotificationManager.class);  
        notificationManager.createNotificationChannel(channel);  
    }  
}
```

Ejemplo (tap action)

```
// Create an explicit intent for an Activity in your app
Intent intent = new Intent(this, AlertDetails.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);

NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle("My notification")
    .setContentText("Hello World!")
    .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    // Set the intent that will fire when the user taps the notification
    .setContentIntent(pendingIntent)
    .setAutoCancel(true);
```

Notificaciones: ¿qué podemos hacer?

MTV: Buscar información sobre cosas que podemos hacer con una notificación

- Sube a la plataforma un documento PDF con la información que hayas encontrado de cosas que se puedan hacer actualmente con una notificación Android.

¿Qué es GCM y FCM? Pasos...

Google Cloud Messaging: servicio de mensajería de Google...pero ha evolucionado y ahora es FCM (Firebase Cloud Messaging). GCM será apagado en Abril de 2019.

Pasos breves:

1. Alta del nuevo proyecto en la Consola de Firebase. Seguir el tutorial para añadir Firebase al proyecto Android. Desde esta consola podremos más adelante genera mensajes para ver que están funcionando en nuestra app. Nos genera un json que tendremos que añadir a nuestro proyecto Android
2. Nos indica las modificaciones de configuración que tenemos que hacer en nuestro proyecto Android
3. Configuramos en nuestro proyecto Android, con API 15 o superior

<https://firebase.google.com/docs/cloud-messaging/android/client?hl=es-419>

Notificaciones: FCM

MTV: Practicar la gestión de notificaciones con Firebase

- Crea una aplicación simple, que se conecta con Firebase y muestra distintos tipos de notificaciones.
- Sube a la plataforma un documento PDF capturas de pantalla de las ventanas de notificaciones, así como el código fuente del fichero principal.