

Best Practises für Push-Notifications **für Android Apps**

Inhaltsverzeichnis

Einleitung.....	2
Was sind Notifications?	3
Einsatzgebiete	4
Guidelines für Notifications.....	5
Was ist zu beachten ist bei der Entwicklung einer App	5
Notification Kanäle	5
Implementierungsbeispiele.....	6
Normale Notification inklusive Action Buttons	6
Messaging Style mit Reply Button.....	6
Fazit	7
Quellen	8

Einleitung

Als Neuling im Thema Android-Apps war es am Anfang schwierig zu finden, wie man einen Benutzer über Vorgänge in seiner App informiert. Eine Google Suche spuckte da schon des Öfteren aus, wie der Benutzer mit seinen Nachrichten umgehen sollte aus, aber nicht wie ein Programmierer diese umsetzen kann. Nach einiger Zeit stieß ich dann auf den Namen Push-Notification. Ab dann, war es nicht mehr schwer schnell Tutorials und andere Anleitungen zu finden. Doch wie sie nun Einsetzen? Man kennt nur all zu gut die Nervigen Pieptöne, wenn mal wieder eine App was Neues hat, der Freund schreibt oder das Spiel gerade dir sagen will, dass deine Basis attackiert wird. Somit war das nächste was ich suchen musste, WIE setze ich diese Nachrichten gezielt ein.

Um anderen diesen langen Weg zu erleichtern, wird innerhalb dieser Arbeit beleuchtet,

- Was Notifications sind
- Wo und wie sie einsetzbar sind
- Welche Richtlinien man folgen sollte
- Welche Zeilen Code benötigt werden

Im Zuge dieser Arbeit ist auch eine Komplette Demo-App entstanden. Diese ist auf GitHub frei verfügbar.

- <https://github.com/Cayleb-Ordo/Notification-Demo>

In dieser Dokument sind nicht alle Notification Arten ausführlich beschrieben. Daher verweise ich an dieser Stelle an dieses Repository und die dort enthaltene README.

Was sind Notifications?

Aufbau

Funktion

Einsatzgebiete

Guidelines für Notifications

Gemäß der Blogseite MobiLoud werden 71% aller App Deinstallationen durch Notifications ausgelöst. Das ist eine recht hohe Anzahl, die recht einfach zu verhindern ist. Ein anderer Blog beschreibt auch, wie diese Dinge einfach umzusetzen sind. CleverTrap schreibt in ihrem Blog, es komme auf drei Wichtige Punkte an.

1. Relevant und zeitgerecht
2. Persönlich
3. Aktionsgerecht

Diese drei Punkte sind die Hauptmerkmale einer Notifications.

Was ist zu beachten ist bei der Entwicklung einer App
Notification Kanäle

Implementierungsbeispiele

Auf der Android-Developer Seite findet man ausführliche Guides, wie Notifications erstellt werden können. Diese sind leider nicht aktuell, manchmal muss man schauen welche der android-Packages benötigt werden. Die hier aufgeführten Codebeispiele sind mit der androidx-Library erstellt. Hier werden exemplarisch zwei Arten von Notification gezeigt. Wichtig ist hierbei das es sich nur um die Notification selbst handelt, nicht darum wie und wann sie eingesetzt wird.

Notification-Kanal

Normale Notification inklusive Action Buttons

Erstellung einer einfachen Push-Notification mit Titel, Tab-Action, Inhalts Text, Icon und einem Action-Button. Es muss bei der Action kein Icon spezifiziert werden. Bei der Unterscheidung der PendingIntent's aufpassen. Diese müssen eindeutig unterscheidbar sein, z. b. eine eindeutige Action.

Code

```
Notification defaultNot = new NotificationCompat.Builder(context,
channelid)
    .setSmallIcon(icon)
    .setContentTitle(contentTitle)
    .setContentText(context.getString(R.string.NotDefContent))
    .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    .setContentIntent(buildContentIntent())
    .addAction(R.drawable.ic_launcher_foreground,
context.getString(R.string.NotActionClose), buildDismissIntent(notID))
    .setAutoCancel(true) // Lässt die Nachricht nicht verschwinden bis
auf sie geklickt wird
    .setGroup(NOTIFICATION_GROUP_KEY)
    .build();
return defaultNot;
```

Messaging Style mit Reply Button

Fazit

Quellen

- <https://www.mobiloud.com/blog/push-notifications-best-practices>
- <https://clevertap.com/blog/push-notification-best-practices/>
-