Docs My Account

Triggers

Understand how to use triggers to create notifications that fire under specific conditions.

← Events Debugging →

Triggers can be used to display notifications in-advance when a specific condition is met such as time.

For example, you may wish to notify your user when they have a meeting at work.

Triggers Example

Creating a trigger notification

```
import React from 'react';
import { View, Button } from 'react-native';
import notifee, { TimestampTrigger, TriggerType } from '@notifee/react-native';

function Screen() {
  async function onCreateTriggerNotification() {
    const date = new Date(Date.now());
    date.setHours(11);
    date.setMinutes(10);

    // Create a time-based trigger
    const trigger: TimestampTrigger = {
      type: TriggerType.TIMESTAMP,
      timestamp: date.getTime(), // fire at 11:10am (10 minutes before meeting)
    };

    // Create a trigger notification
    await notifee.createTriggerNotification(
      {
        title: 'Meeting with Jane',
        body: 'Today at 11:20am',
        android: {
          channelId: 'your-channel-id',
        },
      },
      trigger,
    );
  }

  return (
    <View>
      <Button title="Create Trigger Notification" onPress={() => onCreateTriggerNot
    </View>
  );
}
```

The **`createTriggerNotification`** method is called passing in a **`notification`** and a **`trigger`** . You must ensure the channel is created first as well as other attributes your notification may have such as categories.

Go ahead and press the button! A notification will appear in 10 minutes, giving the user a reminder they have a meeting with Jane!

Updating a trigger notification

Trigger notifications work in the same way as any other notification. They have a random unique ID assigned to them which can be used to update pending trigger notifications. If there is no trigger with that ID, a new one will be created.

Let's update our trigger we created previously to occur weekly.

```
import notifee, { TimestampTrigger, TriggerType } from '@notifee/react-native';

async function onCreateTriggerNotification() {
  const date = new Date(Date.now());
  date.setHours(11);
  date.setMinutes(10);

  const trigger: TimestampTrigger = {
    type: TriggerType.TIMESTAMP,
    timestamp: date.getTime(),
    repeatFrequency: RepeatFrequency.WEEKLY,
  };

  await notifee.createTriggerNotification(
    {
      id: '123',
      title: 'Meeting with Jane',
      body: 'Today at 11:20am',
      android: {
```

```
    trigger,
  );
}
```

> To update any notifications that are already displayed, you can update them using displayNotification

Retrieving trigger notifications

To retrieve a list of all your trigger notifications you can call getTriggerNotificationIds method and a list of ids will be returned.

```
notifee.getTriggerNotificationIds().then(ids => console.log('All trigger notificati
```

Cancelling a trigger notification

There may be situations whereby you may want to cancel the trigger notification and stop any future notifications from displaying.

To cancel a trigger notification, the `cancelNotification` method can be called with the unique notification ID.

It's also possible to cancel all of your trigger notifications, by calling cancelTriggerNotifications or cancelAllNotifications.

Trigger Types

The **`TimestampTrigger`** allows you to create a trigger that displays a notification at a specific time and date, using the property **`timestamp`** and an optional **`repeatFrequency`** property:

```
import notifee, { TimestampTrigger, TriggerType, TimeUnit } from '@notifee/react-nat

const trigger: TimestampTrigger = {
  type: TriggerType.TIMESTAMP,
  timestamp: Date.now() + 1000 * 60 * 60 * 3, // fire in 3 hours
  repeatFrequency: RepeatFrequency.WEEKLY, // repeat once a week
};
```

On Android, you have the option to create your trigger notification with Android's AlarmManger API:

```
const trigger: TimestampTrigger = {
  //...
  alarmManager: true,
};
```

If you want to allow the notification to display when in low-power idle modes, set **`allowWhileIdle`** :

```
const trigger: TimestampTrigger = {
  //...
  alarmManager: {
    allowWhileIdle: true,
  },
};
```

Maximum TimestampTrigger Count

Android has a system limit of 50 timestamp triggers active at one time. iOS appears to have a limit of 64 timestamp triggers active at one time, but it is not in the official documentation.

iOS Initial Trigger Limitations

- **`HOURLY`** : the starting date and hour will be ignored, and only the minutes and seconds will be taken into the account. If the timestamp is set to trigger in 3 hours and repeat every 5th minute of the hour, the alert will not fire in 3 hours, but will instead fire immediately on the next 5th minute of the hour.
- **`DAILY`** : the starting day will be ignored, and only the time will be taken into account. If it is January 1 at 10 AM and you schedule a daily recurring notification for January 2 at 11 AM, it will fire on January 1 at 11 AM and every day thereafter.
- **`WEEKLY`** : the starting week will be ignored, and only the day and time will be taken into account.

> For more details, please see the discussion here.

Android 12 Limitations

Starting from Android 12, timestamp triggers cannot be created unless user specfically allow the exact alarm permission. Before you create a timestamp trigger, check whether **`SCHEDULE_EXACT_ALARM`** permission is allowed by making a call to **`getNotificationSettings`** . If **`alarm`** is **`DISABLED`** , you should educate the user on this permission and ask to enable scheduling alarms. You can then use **`openAlarmPermissionSettings`** function to display the Alarms & Reminder settings of your app.

```
const settings = notifee.getNotificationSettings();
if (settings.android.alarm == AndroidNotificationSetting.ENABLED) {
  //Create timestamp trigger
} else {
  // Show some user information to educate them on what exact alarm permission is,
  // and why it is necessary for your app functionality, then send them to system pr
  await notifee.openAlarmPermissionSettings();
}
```

user and they grant permission again, notifee will automatically reschedule the triggers when the user allows the alarm permission again with no need for additional code.

Interval Trigger

The **IntervalTrigger** allows you to create a trigger that repeats at a specific interval. The trigger accepts two properties, an **interval** and an optional **timeUnit** .

This trigger can be used to implement timers.

For example, to set a trigger to repeat every 30 minutes from now:

```
import notifee, { IntervalTrigger, TriggerType, TimeUnit } from '@notifee/react-nat

const trigger: IntervalTrigger = {
  type: TriggerType.INTERVAL,
  interval: 30,
  timeUnit: TimeUnit.MINUTES
};
```

← Events

Debugging →

GitHub Contact Frequently Asked Questions