

Spec: Notification Stock Alert — แจ้งเตือนอัตโนมัติเมื่อ Stock ต่ำกว่า Minimum

สำหรับ Dev — พร้อม implement

ภาพรวม

เปลี่ยนระบบ notification จาก "แจ้งเฉพาะตอน user กด search ที่หน้า Stock Retrieval" → เป็น "แจ้งอัตโนมัติทันทีที่ stock ต่ำกว่า minimum" โดยเก็บ notification ใน DB แทน localStorage

กฎหลัก

- แจ้ง 1 ครั้งต่อ 1 part จนกว่า stock กลับ \geq minimum
- ไม่แจ้งซ้ำ ตราบใดที่ stock ยังต่ำอยู่
- แจ้งใหม่ได้ ถ้า stock กลับปกติแล้วต่ำอีกรอบ
- Global notification — ทุกคนเห็นเหมือนกัน ไม่ต้อง config role
- ไม่มี seen/unseen mode — กระดิ่งแสดง noti ที่ยัง unresolved ทั้งหมดเสมอ (หายจากกระดิ่งเมื่อ stock กลับปกติเท่านั้น)

1. Database — สร้างตาราง notify_log

```
CREATE TABLE notify_log (  
  id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  notify_category VARCHAR(50) NOT NULL, -- เช่น 'stock_alert'  
  ref_id      VARCHAR(100) NOT NULL, -- part code (ma_part_code)  
  description TEXT,                -- รายละเอียด เช่น "P-BRG-6205 : Bearing 6205 - Stock:  
  3 < Min: 10"  
  is_resolved BOOLEAN DEFAULT false, -- stock กลับปกติแล้วหรือยัง  
  created_date TIMESTAMPTZ DEFAULT NOW()  
);  
  
CREATE INDEX idx_notify_log_unresolved ON notify_log (ref_id, notify_category) WHERE  
is_resolved = false;
```

Prisma model:

```
model notify_log {  
  id          String  @id @default(dbgenerated("gen_random_uuid()")) @db.Uuid  
  notify_category String @db.VarChar(50)  
  ref_id      String  @db.VarChar(100)  
  description String?  
  is_resolved Boolean  @default(false)  
  created_date DateTime @default(now()) @db.Timestamptz()  
}
```

2. Backend — เช็ค Stock หลัง Part Ship

ไฟล์: src/modules/call-monitor/part-ship/part-ship.service.ts

Logic เพิ่มหลัง ship สำเร็จ:

```
// หลังจาก UPDATE part_stock_item.status = 'SHIPPED' สำเร็จแล้ว
async function checkAndNotifyLowStock(maPartId: string) {
  // 1. นับ stock ปัจจุบัน
  const currentStock = await prisma.part_stock_item.count({
    where: {
      part_stock_detail: { ma_part_id: maPartId },
      status: 'STOCK',
    },
  });

  // 2. ดึง minimum_qty
  const part = await prisma.ms_ma_part.findUnique({
    where: { id: maPartId },
    select: { ma_part_code: true, ma_part_name: true, minimum_qty: true },
  });

  if (!part || part.minimum_qty <= 0) return;

  // 3. เช็คต่ำกว่า min มั้ย
  if (currentStock >= part.minimum_qty) return;

  // 4. เช็คว่ามี unresolved alert อยู่แล้วมั้ย
  const existing = await prisma.notify_log.findFirst({
    where: {
      notify_category: 'stock_alert',
      ref_id: part.ma_part_code,
      is_resolved: false,
    },
  });

  if (existing) return; // เคยแจ้งแล้ว ยังไม่ resolved

  // 5. INSERT notify_log
  await prisma.notify_log.create({
    data: {
      notify_category: 'stock_alert',
      ref_id: part.ma_part_code,
      description: `${part.ma_part_code} : ${part.ma_part_name} - Stock: ${currentStock} <
Min: ${part.minimum_qty}`,
      is_resolved: false,
    },
  });
}
```

3. Backend — Resolve ตอน Stock-In

ไฟล์: service ที่ทำ stock-in (Inspection Good / รับของเข้าคลัง)

Logic เพิ่มหลัง stock-in สำเร็จ:

```
// หลังจาก INSERT part_stock_item (status = 'STOCK') สำเร็จแล้ว
async function resolveAlertIfStockRecovered(maPartId: string) {
  const currentStock = await prisma.part_stock_item.count({
    where: {
      part_stock_detail: { ma_part_id: maPartId },
      status: 'STOCK',
    },
  });

  const part = await prisma.ms_ma_part.findUnique({
    where: { id: maPartId },
    select: { ma_part_code: true, minimum_qty: true },
  });

  if (!part) return;

  // ถ้า stock กลับ >= minimum → resolve alert
  if (currentStock >= part.minimum_qty) {
    await prisma.notify_log.updateMany({
      where: {
        notify_category: 'stock_alert',
        ref_id: part.ma_part_code,
        is_resolved: false,
      },
      data: { is_resolved: true },
    });
  }
}
```

4. Backend — API Endpoints

สร้าง module ใหม่: src/modules/notify-log/

GET /api/v1/notify-log/active

ดึง notification ที่ยัง unresolved (สำหรับ Frontend poll แสดงในกระดิ่ง)

```
// Response
{
  data: [
    {
      id: "uuid",
      notify_category: "stock_alert",
      ref_id: "P-BRG-6205",
    }
  ]
}
```

```
    description: "P-BRG-6205 : Bearing 6205 - Stock: 3 < Min: 10",
    created_date: "2026-05-20T10:30:00Z"
  }
],
total: 5
}
```

Logic: ดึง notify_log ที่ is_resolved = false เรียงตาม created_date DESC จำกัด 15 รายการ

POST /api/v1/notify-log/mark-seen

ไม่ต้องทำ — ไม่มี seen mode

5. Frontend — เปลี่ยน useNotifications

ไฟล์: packages/shared/feature/notification/hooks/use-notifications.ts

เปลี่ยนจาก:

- เก็บ noti ใน localStorage
- เพิ่ม noti เฉพาะตอน user กด search

เป็น:

- Poll API GET /api/v1/notify-log/active ทุก 60 วินาที
- แสดง noti ที่ยัง unresolved ทั้งหมดในกระดิ่ง (ไม่มี seen/unseen)
- Badge = จำนวน noti ที่ยัง unresolved
- noti หายจากกระดิ่งเมื่อ stock กลับปกติ (is_resolved = true) เท่านั้น
- เสียง 🔔 ดังเมื่อมี noti ใหม่ (เทียบ count กับรอบก่อน)

```
// Pseudo-code
const { data } = useQuery({
  queryKey: ['notify-log-active'],
  queryFn: () => fetch('/api/v1/notify-log/active'),
  refetchInterval: 60_000, // poll ทุก 1 นาที
});

const badgeCount = data?.total ?? 0; // แสดงจำนวน unresolved ทั้งหมด
```

6. Frontend — กระดิ่ง (NotificationBell)

ไม่เปลี่ยน UI — ยังใช้ popover + badge เหมือนเดิม

เปลี่ยนแค่ data source + behavior:

- Badge = จำนวน noti ที่ is_resolved = false (unresolved)
- Popover แสดง noti จาก API (ไม่เกิน 15 ตัว)
- ไม่มี seen mode — เปิดกระดิ่ง badge ไม่หาย (ยังแสดงจำนวน unresolved อยู่)
- noti หายจาก popover เมื่อ stock กลับปกติเท่านั้น (backend resolve)

7. สิ่งที่ไม่ต้องแก้

- ❌ แถบแดงในตาราง Stock Retrieval — ยังใช้ logic เดิม
- ❌ เสียง notification — ยังใช้ Web Audio API เดิม
- ❌ ไม่ต้องเพิ่ม role/permission
- ❌ ไม่ต้องทำ Cron Job (stock เปลี่ยนผ่าน Part Ship เท่านั้น)
- ❌ ลบ notifyLowStockParts() ใน useStockRetrievalLocation.ts ออกได้ (เพราะ backend จัดการแทนแล้ว)

8. ไฟล์ที่ต้องแก้/สร้าง

ไฟล์	Action	รายละเอียด
prisma/schema.prisma	แก้	เพิ่ม model notify_log
src/modules/notify-log/	สร้างใหม่	controller, service, repository
src/modules/call-monitor/part-ship/part-ship.service.ts	แก้	เพิ่ม checkAndNotifyLowStock() หลัง ship
service ที่ทำ stock-in	แก้	เพิ่ม resolveAlertIfStockRecovered() หลัง stock-in
packages/shared/feature/notification/hooks/use-notifications.ts	แก้	เปลี่ยนจาก localStorage → poll API
packages/shared/feature/stock-retrieval/hooks/useStockRetrievalLocation.ts	แก้	ลบ notifyLowStockParts()

9. Seed Query — รันครั้งแรกตอน Deploy

รันครั้งเดียวหลัง deploy เพื่อ INSERT notify_log สำหรับ part ที่ stock ต่ำกว่า min อยู่แล้ว ณ ปัจจุบัน (เพราะ part เหล่านี้ไม่เคยผ่าน Part Ship flow ใหม่ จึงยังไม่มี record ใน notify_log)

```
INSERT INTO notify_log (notify_category, ref_id, description, is_resolved, created_date)
SELECT
  'stock_alert',
  p.ma_part_code,
  p.ma_part_code || ' : ' || p.ma_part_name || ' - Stock: ' || COALESCE(stock.current_stock,
0) || ' < Min: ' || p.minimum_qty,
  false,
  NOW()
FROM ms_ma_part p
LEFT JOIN (
  SELECT
    psd.ma_part_id,
    COUNT(psi.id) AS current_stock
  FROM part_stock_detail psd
  JOIN part_stock_item psi ON psi.part_stock_detail_id = psd.id
  WHERE psi.status = 'STOCK'
```

```

GROUP BY psd.ma_part_id
) stock ON stock.ma_part_id = p.id
WHERE p.is_active = true
AND p.is_deleted = false
AND p.minimum_qty > 0
AND COALESCE(stock.current_stock, 0) < p.minimum_qty
-- ไม่ INSERT ซ้ำถ้ามี record อยู่แล้ว (กรณีรัน query ซ้ำ)
AND NOT EXISTS (
SELECT 1 FROM notify_log n1
WHERE n1.ref_id = p.ma_part_code
AND n1.notify_category = 'stock_alert'
AND n1.is_resolved = false
);

```

หมายเหตุ: รัน query นี้ 1 ครั้งหลัง deploy ทีเลอร์ใหม่ หลังจากนั้น Part Ship flow จะจัดการ INSERT ให้อัตโนมัติ

10. Flow Diagram

Part Ship (เบิกอะไหล่)

```

|
|--- UPDATE part_stock_item.status = 'SHIPPED'
|
|--- นับ stock ที่เหลือ (status = 'STOCK')
|
|--- stock < minimum_qty?
|   |--- NO → จบ
|   |--- YES → มี unresolved alert อยู่แล้ว?
|       |--- YES → จบ (ไม่แจ้งซ้ำ)
|       |--- NO → INSERT notify_log → กระดิ่ง 📢
|

```

Stock-In (รับอะไหล่เข้าคลัง)

```

|
|--- INSERT part_stock_item (status = 'STOCK')
|
|--- นับ stock ใหม่
|
|--- stock >= minimum_qty?
|   |--- NO → จบ
|   |--- YES → UPDATE notify_log SET is_resolved = true
|

```

Frontend (ทุก 1 นาที)

```

|
|--- GET /api/v1/notify-log/unseen
|
|--- มี noti ใหม่ (created_date > lastSeenDate)?
|   |--- NO → จบ
|   |--- YES → badge +N + เสียง 📢
|

```