**Notification Service**

* **Entity Design:**
  + A Notification entity to store the data (e.g., user ID, type, message, timestamp, status).
  + Example fields:

java

Copy code

public class Notification {

private Long id;

private Long userId;

private String type; // e.g., "Order Placed", "Promotion"

private String message;

private String status; // e.g., "Pending", "Sent"

private LocalDateTime createdAt;

}

* **Cron Jobs:**
  + Use Spring's @Scheduled annotation to periodically check for unsent notifications and send them.
  + Example:

java

Copy code

@Scheduled(fixedRate = 60000) // Runs every 60 seconds

public void processPendingNotifications() {

List<Notification> pendingNotifications = notificationRepository.findByStatus("Pending");

for (Notification notification : pendingNotifications) {

// Send notification logic

notification.setStatus("Sent");

notificationRepository.save(notification);

}

}

* **Integration:**
  + Use REST calls or a shared resource server for secure communication with the product and user services.
  + Example use case: Fetch user details from the user service for personalized notifications.