

Seguimiento de Tareas Asincrónicas en Tiempo Real

En sistemas modernos, muchas operaciones pueden requerir tiempo considerable para completarse: generación de reportes, carga de archivos, sincronización de datos, entre otras. En esta práctica, se desarrollará un mecanismo que permita a los usuarios monitorear el avance de dichas tareas en tiempo real, utilizando WebSockets como canal de actualización.

Diseñar e implementar un sistema de seguimiento de tareas asincrónicas en backend que informe al usuario, en tiempo real, sobre el estado y progreso de una tarea iniciada por él mismo.

Requerimientos Funcionales

1. Entidad TareaProceso:

Se debe modelar e implementar una entidad en base de datos con los siguientes campos mínimos:

- id: Identificador único de la tarea.
- usuarioid: Usuario que solicitó la tarea.
- tipo: Descripción del tipo de tarea (ej. "generar reporte").
- estado: Puede ser pendiente, en-proceso, completado.
- progreso: Número entero entre 0 y 100.
- resultado: Texto opcional con el resultado o mensaje final.
- fechaInicio y fechaFin.

2. Inicio de Tarea:

- Exponer un endpoint tipo POST /tareas/iniciar.
- Este endpoint debe crear una nueva instancia de TareaProceso y **simular una tarea asincrónica** que avance en el tiempo.
- Durante la simulación, se debe:
 - Actualizar el progreso en la base de datos.
 - Emitir cada avance vía WebSocket al usuario autenticado.
- Simulación del progreso:
El avance de la tarea será simulado utilizando intervalos de tiempo programados. Por ejemplo, se puede utilizar `setInterval` para incrementar el campo `progreso` de la tarea cada cierto tiempo (ej. cada 5 o 10 segundos y que incremente de 20%), hasta alcanzar el 100%.
Esta simulación representa de forma artificial el comportamiento de procesos

reales que se ejecutan en segundo plano. Cada vez que el progreso aumente, se debe actualizar en la base de datos y emitir un evento WebSocket correspondiente al usuario autenticado.

3. WebSocket Gateway de Seguimiento:

- Crear un módulo **ProgresoModule** y un gateway **ProgresoGateway**.
- Solo usuarios autenticados mediante JWT podrán conectarse.
- Emitir eventos al usuario con el estado de la tarea. Ejemplo:

```
{
  "tareaId": 42,
  "progreso": 60,
  "mensaje": "Procesando datos intermedios"
}
```

Indicaciones Adicionales

- No se requiere desarrollar una interfaz con diseño elaborado. Basta con mostrar el progreso mediante logs o un HTML básico.
- Asegúrese de que solo el usuario que inició la tarea reciba el progreso correspondiente.
- La estructura del proyecto debe seguir buenas prácticas de modularidad en NestJS.