

Arquitectura de software 2

Alumno: Santiago Vietto

Docente: Eduardo Carlos Gaite

DNI: 42654882

Institución: UCC

Año: 2021

Análisis funciones

Consigna

_ Escribir un Servicio Web, que permita tomar los datos de una red de sensores de temperatura (Simulado) (**Temperatura**, **presión** Atmosférica, **Velocidad** del Viento). Describir el Protocolo con los datos que tienen que enviar los sensores. Al ser una red cada sensor tiene una ubicación. Y pueden estar **Activados o Desactivados**.

ENTONCES:

- Simular que los sensores envían información una vez **por hora**. (24 registros diarios)
- Tener un método para poder acceder a las últimas 24 mediciones de un sensor.
- Tener un método para saber todos los sensores habilitados en una **ubicación** (listo)
- Habilitar un método para poder activar un sensor.

Tablas:

Sensores:

Sensor_ID	Ubicación_ID	On/Off
1	1	
2	1	

Registros:

Registro_ID	Temperatura	Presión Atmosférica	Velocidad	Timestamp	Sensor_ID
1	234	321	123	1/1 13:0	2

Ubicaciones:

ID	Nombre
1	Mendoza

Análisis funciones

1)_ Simular que los sensores envían información una vez por hora. (24 registros diarios):

_ Para simular que manda datos lo que hay que usar es un POST, ya que este sirve para agregar cosas a la base de datos. El siguiente comando es el que hay que poner en una terminal para hacer un POST, y lo hacemos en la dirección donde esta tomcat, y lo hacemos en /register porque así está definido para registrar un sensor en el controlador del registro:

- `curl -i -H 'Content-Type: application/json' -X POST -d '{"temperature":"$tmp", "pressure":"$prs"."$prss", "wind":"$wnd", "timestamp":"2022-02-20T"$hour":00:00.000+0000", "sensor_id":"$sensor"}'`
`http://localhost:8080/service/register`

_ Este archivo lo que hace es mediante un for loop, que lo que va a hacer es entre para cada hora entre 0 y 23 (es decir, 00, 01, 02, 03, etc), y para cada uno de los sensores con ID del 1 al 10, va a generar un POST mediante un curl con valores random para la temperatura, presión y la velocidad del viento, y así terminarían siendo como 2400 POST para simular que cada sensor hace una medición por hora. Tenemos dos valores de presión porque uno es para valores enteros y el segundo representa la misma presión pero el valor en decimal. Lo que está en \$ se reemplaza con la variable que se generó más arriba.

2)_ Tener un método para poder acceder a las últimas 24 mediciones de un sensor. (getMeditions()):

_ SimpleDateFormat es para mejorar la presentación de la fecha cuando aparece, en donde el objeto sdf que va a poder formatear un dato en milisegundos a una fecha que sea en año/mes/día/hora/minutos/segundos.

_ Buscamos los registros que hay en la base de datos, cargamos una lista de registros y creamos la lista de respuestas de DTOS que va a ser lo que vamos a mostrar al último. Luego tenemos el objeto DTO que vamos a reutilizar para ir cargando en la lista aquellos registros que cumplen con el requisito de haberse realizado en las últimas 24 horas. Con respecto a la lógica, hacemos la matemática en milisegundos en donde tenemos 24 horas llevado a milisegundos. Entonces, en el ciclo for, para cada registro que encontramos en la base de datos, nos fijamos la diferencia de tiempo que tienen, desde el tiempo actual en el que hacemos la consulta menos el tiempo en el que se

haya realizado el registro que buscamos en la base de datos; ahora si la diferencia de tiempo es menor a dayInMili quiere decir que el registro se realizo en las ultimas 24 horas, si la diferencia es mayor significa que fue hace mas de 24 horas por lo que no se carga. Entonces si el sensor está dentro de las ultimas 24 horas cargamos el DTO de respuesta con los datos y lo añadimos a la lista de DTOS para luego devolverlos, en caso contrario no lo toma y sigue con el siguiente sensor.

_ Para que sea más eficiente el método debería filtrarse antes por SQL en el DAO, en lugar de hacerlo en el servicio, para directamente buscar lo que necesitamos de la base de datos en lugar de buscar todo y después filtrar.

3)_ Tener un método para saber todos los sensores habilitados en una ubicación:

_ Traemos los sensores del DAO, creamos la lista de DTOS que va a devolver, creamos el objeto DTO, por el HTTP pasamos un ORDINAL, es decir, un 0, 1, 2 o 3 en donde cada uno hace referencia a una ubicación, y guardamos el valor que estamos buscando en una variable location. Por cada uno de los sensores que buscamos en la base de datos, creamos un nuevo DTO de respuesta, y si el sensor esta prendido y el lugar en donde esta el sensor es igual al que pasamos por parámetro, creamos el DTO agregando sus datos y lo ponemos en la nueva lista que vamos a devolver, y una vez que revisamos todos los sensores para ver cuales están prendidos en el lugar donde estamos consultando, armo la lista orginial de DTOS y los devuelvo.

4)_ Habilitar un método para poder activar un sensor.

_ Lo primero que hace el método es crear un objeto sensor, cargando a través de un DAO de la base de datos según el ID, es decir, si buscamos el sensor 7 lo primero que hacemos es buscarlo en la base de datos y lo guardamos como DAO. Luego le preguntamos al sensor que obtuvimos si su estado es APAGADO, en caso de que si le seteamos el estado ENCENDIDO, ahora, en caso de que el estado del sensor no sea APAGADO, esto quiere decir que ya está encendido por lo que se lo apaga. Al ser un toogle prende o apaga el sensor dependiendo de su estado. Con update actualizamos el mismo sensor en la base de datos, ya que estamos cambiando su estado. Creamos un DTO de respuesta que va a ser el que se va a mostrar por pantalla y le seteamos los datos que va a mostrar, como el estado (que lo pasamos a string), le seteamos el nombre de la locación del sensor (que lo pasamos a string), y por último le seteamos el ID. El return se hace cargo el controlador.

Aclaraciones:

_ Hacer un PUT que permita habilitar o deshabilitar un sensor, para hacer un update en la base de datos.

_ Un listado de las últimas mediciones del sensor.

_ Un POST en el cual el sensor hace un registro de la temperatura, presión atmosférica y demás. En este POST que se verifique que el sensor este habilitado, porque si esta deshabilitado que no registre nada y devuelva un 404.