



Problema Lab 06

Java

CREA UNA TABLA DE USUARIOS Y UN ENDPOINT PARA PODER AUTENTICARLOS

En este lab vamos a aprender a trabajar la autorización y la autenticación de usuarios dentro del contexto de las aplicaciones y las API REST. Gracias a esto podremos limitar el acceso a los recursos que exponemos a internet para que solo los usuarios que nosotros elijamos puedan acceder a según qué elementos o áreas.

Antes de empezar a leer el problema, asegúrate de que controlas los siguientes conceptos que hemos visto a lo largo de esta semana y estudiado el fastbook correspondiente.

- Creación y manejo de usuarios.
- Asignación de roles.
- Configuración del proyecto para utilizar la autenticación, autorización y poder generar JWTs.
- Saber aplicar métodos de seguridad a cada *endpoint* para un rol específico.

Objetivos de este ejercicio

La función de este reto es que seas capaz de ampliar el ejercicio del lab anterior y sepas configurar la parte de autorización y de autenticación a tu aplicación en Java. También buscamos que intentes razonar qué estructura o pasos debes seguir para alcanzar el objetivo propuesto.

Descripción de la actividad

Queremos que sigas con el proyecto visto en el laboratorio anterior, ampliando la funcionalidad que ya tienes implementada dentro de tu aplicación. Para ello, deberás trabajar la configuración necesaria para poder utilizar la autenticación y autorización en Spring Boot.

En primer lugar, debes aplicar la autenticación para poder generar *tokens* y, después, la autorización para que, dado ese *token*, un usuario pueda acceder a los recursos o *endpoints* que se han definido.

A continuación, te compartimos las condiciones generales que debes tener en cuenta durante el proceso de trabajo. ¡Toma nota!

- Debes configurar el proyecto para que este pueda usar la autenticación y la autorización. Además, vas a inicializar 2 propiedades.
 - El *secret-key* de la aplicación para firmar el JWT.
 - El tiempo de expiración del JWT.

- Añade una tabla *users* e inserta usuarios en la aplicación. Esta tabla contendrá los campos: *id*, *name*, *mail*, *password*, *role* (todos serán de tipo *varchar* salvo el *id*, que será de tipo *int*, *primary key* y autoincremental). También deberás codificar la contraseña al guardarla en la BD.

Nota: si quieres crear una tabla maestra de roles en la que se almacenen los posibles roles de la aplicación, lo dejamos a tu elección. Ten en cuenta que, a la hora de asignarlo al usuario, ha de recuperar solo el valor y no el *id*.

- Trabaja la configuración necesaria para poder añadir y *loguear* usuarios en la aplicación. Estos *endpoints* tienen que devolver automáticamente el JWT para poder acceder a la aplicación. Por defecto, cuando crees un nuevo usuario, le asignarás el rol *ROLE_CLIENT*.
- Incluye la securización a los endpoints que ya tenías creados con anterioridad teniendo en cuenta que:
 - los métodos GET solo podrán ser consultados por los usuarios con *ROLE_CLIENT*;
 - los métodos POST, PUT, DELETE solo podrán ser consultados por los usuarios con *ROLE_VENDOR*.

Formato de entrega

Envía tu ejercicio en un archivo con extensión .java o, si has utilizado más de una clase para resolver la actividad, un archivo comprimido en formato .zip o .rar con el conjunto de clases empleadas durante la resolución del problema.

Criterios de corrección

Te compartimos algunos puntos que debes comprobar sí o sí cuando vayas a visualizar la solución del profesor y a autoevaluar tu ejercicio antes de darlo por superado.

- Primero, tienes que comprobar que la aplicación funciona de manera correcta. Para ello, debes comprobar que muestra el mensaje de inicio al arrancar en el puerto 8080 y que no hay fallos.
- Segundos, debes comprobar que los datos que se muestran son los adecuados.
- Tercero, incluye comentarios sobre cada funcionalidad para comprobar que entiendes realmente lo que hace. Estos comentarios deben explicar cada instrucción.

Si has sido capaz de implementar todos estos puntos con éxito, ¡enhorabuena! Ya sabes aplicar la autorización y autenticación de usuarios en tu aplicación en Java utilizando Spring Boot. ¡Reto superado!



Qualentum.com