

Challenge Backend

Deberás desarrollar una API para un blog, utilizando java + Spring boot. Los datos deberán ser persistidos en una base de datos Postgres o MySQL. Esta API deberá devolver datos en formato JSON.

Primera iteración

Deberá poder registrar un usuario, y luego poder loggarse.

El usuario deberá tener los siguientes campos

- ID
- email
- password

Se debe usar Spring Security para la autenticación, pudiendo cambiar los campos del usuario. Preferentemente autenticar utilizando JWT.

Cada post deberá contener los siguientes campos:

- ID
- Título
- Contenido
- Imagen
- Categoría
- Fecha de creación
- ID del usuario creador

Los endpoints que deberá exponer la API son:

Los endpoints de autenticación podrían cambiar según la gema utilizada, pero para utilizar de guía se sugiere:

POST /auth/sign_up

Se debe enviar email y contraseña y se creara un nuevo usuario con esos datos.

POST /auth/login

Se debe enviar usuario y contraseña para luego poder utilizar cualquiera de los endpoints siguientes

GET /posts

Deberá mostrar un listado de posts, ordenados por fecha de creación, en forma descendente. Este listado deberá mostrar solamente los campos ID, título, imagen, categoría y fecha de creación.

Se deberá poder filtrar por título y/o categoría.

/posts?title=TITULO

/posts?category=CATEGORIA

/posts?titulo=TITULO&category=CATEGORY

GET /posts/:id

Deberá buscar un post cuyo id sea el especificado en el parámetro :id. Si existe, devolver sus detalles, caso contrario devolver un mensaje de error.

POST /posts

Deberá guardar un nuevo post según los datos recibidos en la petición.

PATCH /posts/:id

Deberá actualizar el post con el id especificado en el parámetro :id, y actualizar sus datos según el cuerpo de la petición. En el caso de que no exista, devolver un mensaje de error.

DELETE /posts/:id

Deberá eliminar el post con el id especificado en el parámetro :id. En el caso de que no exista, devolver un mensaje de error.

Consideraciones

Se evaluará la correcta estructuración de los archivos, buena escritura del código.

De forma adicional, puede crearse una tabla adicional para las categorías y relacionar las mismas a través de una clave foránea en cada post.

El campo imagen debe ser la URL de una imagen, no es necesario estructurar una lógica de almacenamiento de archivos, puede ser una referencia hacia un sitio externo (es un bonus validar que la URL corresponda una imagen, por ejemplo, que finalice en .png, jpg, o comprobar la existencia del recurso antes de almacenarlo)

Segunda iteración

Se deberá cambiar la lógica de borrado de posts, para implementar un soft delete o borrado lógico.

Investigar qué significa esto, y de qué manera se puede implementar e implementarlo.

Tercera iteración (opcional)

Se deberán agregar tests para los métodos de los controladores.

Se tiene que usar Junit y Mockito.

En cualquier caso, además de los tests, escribir un breve detalle con comentarios de que casos se están testeando.