

Challenge Backend

Deberás desarrollar una API para un blog, utilizando Node y Express. Los datos deberán ser persistidos en una base de datos MySQL.

Cada post deberá contener los siguientes campos:

- ID
- Título
- Contenido
- Imagen
- Categoría
- Fecha de creación

Los endpoints que deberá exponer la API son:

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.

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. Es deseable que se utilice un ORM (por ej. Sequelize)

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

Challenge Frontend

Deberás desarrollar un cliente para un Blog.

El mismo debe ser armado en React, y consumir los datos de una [JSON Placeholder](#), una API que expone datos ficticios en formato JSON.

Los endpoints que deberás utilizar son:

GET <https://jsonplaceholder.typicode.com/posts>

Devuelve un listado de posts

GET <https://jsonplaceholder.typicode.com/posts/:id>

Devuelve el detalle de un post en base al id especificado en el parámetro

POST <https://jsonplaceholder.typicode.com/posts>

Simula la creación de un nuevo post

PUT/PATCH <https://jsonplaceholder.typicode.com/posts/:id>

Simula la actualización de un post en base al id especificado en el parámetro

DELETE <https://jsonplaceholder.typicode.com/posts/:id>

Simula la eliminación de un post en base al id especificado en el parámetro

Utilizando estos endpoints, la aplicación deberá contener las siguientes secciones:

Home

Mostrará un listado de posts. En este listado, deberá mostrarse solamente el título de cada uno, y las acciones para ir al detalle del mismo, editarlo o eliminarlo.

Detalle

Deberá recibir el identificador de un post y, en el caso de que exista, mostrar sus datos. Caso contrario, deberá mostrar un mensaje de error.

Formulario de creación

Deberá mostrar un formulario que permita crear un nuevo post. El formulario deberá contener los campos título y contenido, y realizar la validación de los mismos (ambos son obligatorios). Al hacer el submit, debe realizarse la petición al endpoint correspondiente.

Formulario de Edición

Deberá recibir el identificador de un post y mostrar un formulario que permita editarlo. En el caso de que no exista, mostrar un mensaje de error. El formulario deberá contener los campos título y contenido, y realizar la validación de los mismos (ambos son obligatorios). Al hacer el submit, debe realizarse la petición al endpoint correspondiente.

Otras consideraciones

La app deberá contener un encabezado con los links al Home y al Formulario de Edición.
Debe ser responsive, se puede utilizar una plantilla.

La acción de “Eliminar” que contendrán los posts listados en la sección Home deberán realizar la petición al endpoint correspondiente.