CRUD EXPRESS

Objetivo

Demonstre como fazer as quatro operações de CRUD e suas rotas para obter o recurso de um livro que está no estoque de uma livraria. Um livro possui informações como nome, autor e ISBN.

CREATE

router.post('/book/add', async (req, res) => {

const { titulo, subTitulo, autor, genero, ano, isbn } = req.body;

await newBook(titulo, subTitulo, autor, genero, ano, isbn);

res.status(201).send("Livro cadastrado com sucesso!");

});

O endpoint do bloco de códigos acima, envia por parâmetros obtidos do body os dados necessários para a criação de um livro em sua tabela, ao ser criado é retornado um status code 201.

READ

app.get('/book', async (req, res) => {

const mostrarLivros = await allBooks();

if (!mostrarLivros) {

res.sendStatus(204);

} else {

res.status(200).send(mostrarLivros);

}

});

O endpoint do bloco de códigos acima, realiza um SELECT \* da tabela dentro do banco de dados para retornar a leitura de todos os livros cadastrados neste DB.

UPDATE

router.put('/book/editar/:id', async (req, res) => {

let id = req.params.id;

const { genero } = req.body;

const editar = await upBookGen(id, genero);

if (id > 0 && id <= editar){

res.status(200).send("Gênero editado com sucesso!");

} else {

res.status(404);

}

});

O endpoint do bloco de códigos acima, realiza um UPDATE no livro do ID passado no parâmetro da rota, e no corpo será informado qual é o gênero a ser editado do livro selecionado pelo seu id, caso seja passado id 0 ou inexistente, é retornado um error 404 not found.

DELETE

router.delete('/book/deletar/:id', async (req, res) => {

let id = req.params.id;

const deletar = await delBook(id);

if (id > 0 && id <= deletar) {

res.status(200).send("Livro removido com sucesso!");

} else {

res.status(404);

}

O endpoint do bloco de códigos acima, realiza um DELETE no livro do ID passado no parâmetro da rota, caso seja passado id 0 ou inexistente, é retornado um error 404 not found. Caso o id passado esteja correto o registro no DB que possui tal id, será excluído da base de dados.

Observações Finais

Os endpoints que possuem router como: router.post, router.put e router.delete, respectivamente responsáveis pelo CREATE, UPDATE E DELETE de registros no DB, estão em outro diretório administrativo denominado como router, tendo em vista que um client não poderá realizar tais requisições, apenas o app.get responsável pela leitura do conteúdo.