

Lista III - Desenvolvimento de uma API REST para Gestão de Biblioteca

Objetivo

Desenvolver uma API em TypeScript utilizando Node.js, Express e MySQL, que permita o cadastro, consulta, atualização e exclusão de livros em uma biblioteca. A API deve seguir o padrão de projeto MVC e manipular os dados em formato JSON.

Descrição

Você deverá criar uma API que gerencie um acervo de livros. Para isso a mesma deverá permitir as seguintes operações:

- Criar um novo livro (Create)
- Consultar os livros cadastrados (Read)
- Consultar um livro por ID (Read)
- Atualizar informações de um livro existente (Update)
- Excluir um livro (Delete)

Especificações dos Endpoints

1. Criar um Novo Livro

- **Método HTTP:** POST
- **Endpoint:** /books
- **Descrição:** Este endpoint deve permitir a criação de um novo livro na biblioteca. Deve verificar se o livro já existe no acervo pelo campo **isbn**.
- **Contrato JSON de Requisição:**

```
{  
  "title": "string",  
  "author": "string",  
  "publishedDate": "string",  
  "isbn": "string",  
  "pages": "number",  
  "language": "string",  
  "publisher": "string"  
}
```

- **Contrato JSON de Resposta:**

```
{
    "id": "number",
    "title": "string",
    "author": "string",
    "publishedDate": "string",
    "isbn": "string",
    "pages": "number",
    "language": "string",
    "publisher": "string"
}
```

2. Consultar Todos os Livros

- **Método HTTP:** GET
- **Endpoint:** /books
- **Descrição:** Este endpoint deve permitir a consulta de todos os livros cadastrados na biblioteca.
- **Contrato JSON de Resposta:**

```
[
    {
        "id": "number",
        "title": "string",
        "author": "string",
        "publishedDate": "string",
        "isbn": "string",
        "pages": "number",
        "language": "string",
        "publisher": "string"
    }
]
```

3. Consultar um Livro por ID

- **Método HTTP:** GET
- **Endpoint:** /books/:id
- **Descrição:** Este endpoint deve permitir a consulta de um livro específico pelo seu ID. Deve verificar se o livro com o ID fornecido existe.
- **Contrato JSON de Resposta:**

```
{
    "id": "number",
    "title": "string",
    "author": "string",
    "publishedDate": "string",
}
```

```
        "isbn": "string",  
        "pages": "number",  
        "language": "string",  
        "publisher": "string"  
    }  
}
```

4. Atualizar um Livro

- **Método HTTP:** PUT
- **Endpoint:** /books/:id
- **Descrição:** Este endpoint deve permitir a atualização das informações de um livro específico pelo seu ID. Deve verificar se o livro com o ID fornecido existe e se os campos atualizados são válidos.
- **Contrato JSON de Requisição:**

```
{  
    "title": "string",  
    "author": "string",  
    "publishedDate": "string",  
    "isbn": "string",  
    "pages": "number",  
    "language": "string",  
    "publisher": "string"  
}
```

- **Contrato JSON de Resposta:**

```
{  
    "id": "number",  
    "title": "string",  
    "author": "string",  
    "publishedDate": "string",  
    "isbn": "string",  
    "pages": "number",  
    "language": "string",  
    "publisher": "string"  
}
```

5. Excluir um Livro

- **Método HTTP:** DELETE
- **Endpoint:** /books/:id
- **Descrição:** Este endpoint deve permitir a exclusão de um livro específico pelo seu ID. Deve verificar se o livro com o ID fornecido existe.
- **Contrato JSON de Resposta:**

```
{  
    "message": "Book deleted successfully"  
}
```

Requisitos Adicionais

1. **Validação:** Implementar validações para garantir que todos os campos obrigatórios estão sendo enviados e possuem valores válidos. Verificar se o **isbn** é único ao criar um novo livro.
2. **Tratamento de Erros:** A API deve retornar respostas apropriadas em caso de erros, como **400 Bad Request** para requisições inválidas, **404 Not Found** para livros não encontrados e **409 Conflict** para tentativas de criação de livros com **isbn** duplicado.
3. **Persistência de Dados:** Utilizar o banco de dados MySQL.

Entrega

O código-fonte deve ser entregue em um repositório Git. O envio do mesmo se dará via Moodle. O prazo de entrega é até a data **17/07/2024**.

Avaliação

Os critérios de avaliação incluirão:

- **Funcionalidade:** Todos os endpoints funcionam conforme especificado.
- **Código:** Clareza, organização e boas práticas de programação.
- **Validação e Tratamento de Erros:** Implementação adequada de validação de dados e tratamento de erros.