

Exercício de Teste de API - Livros

Objetivo: Este exercício tem como objetivo familiarizar os participantes com testes de API RESTful usando a API de Livros. Você irá realizar testes de Listagem, Consulta, Criação, Atualização e Exclusão (CRUD) de livros, validando códigos de status HTTP e o conteúdo das respostas.

Endpoint Base: `http://localhost:3000/`

Endpoints a serem utilizados:

- **Listar todos os livros:** GET `/livros`
 - **Obter detalhes de um livro:** GET `/livros/{id}`
 - **Criar um novo livro:** POST `/livros`
 - **Atualizar um livro existente:** PUT `/livros/{id}`
 - **Excluir um livro:** DELETE `/livros/{id}`
-

Instruções:

1. Teste de Listagem (GET `/livros`)

- **Objetivo:** Obter a lista de todos os livros cadastrados.
- **Passos:**
 1. Execute uma solicitação **GET** para `/livros`.
 2. Verifique se a resposta tem o código de status **200 OK**.
 3. Analise o formato da resposta para garantir que todos os campos esperados (id, nome, autor, paginas) estejam presentes.

2. Teste de Consulta (GET `/livros/{id}`)

- **Objetivo:** Obter detalhes de um livro específico.
- **Passos:**
 1. Escolha um id de um livro existente.
 2. Execute uma solicitação **GET** para `/livros/{id}` (substitua `{id}` pelo ID escolhido).
 3. Verifique se a resposta tem o código de status **200 OK**.
 4. Confirme se os detalhes do livro correspondem aos dados esperados.

3. Teste de Criação (POST `/livros`)

- **Objetivo:** Criar um novo livro no sistema.
- **Passos:**
 1. Execute uma solicitação **POST** para `/livros`.

2. No corpo da requisição, envie os dados do livro em formato JSON:

```
{  
  "nome": "Título do Livro",  
  "autor": "Nome do Autor",  
  "paginas": 250  
}
```

3. Verifique se a resposta tem o código de status **201 Created**.
4. Confirme se a resposta contém o livro criado com um id atribuído.

4. Teste de Atualização (PUT /livros/{id})

- **Objetivo:** Atualizar as informações de um livro existente.
- **Passos:**
 1. Escolha um id de um livro existente.
 2. Execute uma solicitação **PUT** para /livros/{id}.
 3. No corpo da requisição, envie os campos que deseja atualizar:

```
{  
  "nome": "Novo Título do Livro",  
  "paginas": 300  
}
```

4. Verifique se a resposta tem o código de status **200 OK**.
5. Confirme se os dados do livro foram atualizados corretamente.

5. Teste de Exclusão (DELETE /livros/{id})

- **Objetivo:** Excluir um livro do sistema.
- **Passos:**
 1. Escolha um id de um livro existente.
 2. Execute uma solicitação **DELETE** para /livros/{id}.
 3. Verifique se a resposta tem o código de status **204 No Content**.
 4. Tente consultar o livro excluído usando **GET /livros/{id}** e verifique se a resposta é um código de status **404 Not Found**.

Validações Adicionais

6. Teste de Erro ao Consultar Livro Inexistente

- **Objetivo:** Verificar o comportamento da API ao solicitar um livro que não existe.
- **Passos:**
 1. Escolha um id que não existe no sistema.
 2. Execute uma solicitação **GET** para /livros/{id_inexistente}.
 3. Verifique se a resposta tem o código de status **404 Not Found**.
 4. Confirme se a resposta contém uma mensagem de erro apropriada, como "Livro não encontrado".

7. Teste de Erro ao Criar Livro com Dados Incompletos

- **Objetivo:** Testar a validação de dados na criação de um novo livro.
- **Passos:**
 1. Execute uma solicitação **POST** para /livros.
 2. No corpo da requisição, envie dados incompletos (por exemplo, sem o campo nome):

```
{  
  "autor": "Autor Sem Nome",  
  "paginas": 200  
}
```

3. Verifique se a resposta tem o código de status **400 Bad Request**.
4. Confirme se a resposta contém uma mensagem de erro indicando quais campos estão faltando.