



Disciplina: **Desenvolvimento para Ambientes Corporativos**

Nome: Cristhian Rodrigues Kindermann

GRR: 20214914

Documentação Detalhada da API de Biblioteca

Introdução

Esta documentação descreve os endpoints disponíveis na API de Biblioteca, os formatos de requisição e resposta, e os códigos de status utilizados.

- **URL Base:** `http://localhost:8080`
- **Formato de Dados:** `application/json`

Formato Padrão de Erro

Quando uma requisição falha (status 4xx ou 5xx), a API retorna um corpo JSON com a seguinte estrutura:

```
{
  "timestamp": "2025-08-30T20:05:00.12345Z",
  "codigo": 404,
  "erro": "Recurso não encontrado",
  "caminho": "/livros/999",
  "mensagem": "Livro não encontrado com id: 999"
}
```

Recurso: Livros (`/livros`)

1. `GET /livros`

Lista todos os livros de forma paginada.

- **Query Parameters (Opcionais):**
 - `page` (int): O número da página a ser retornada (padrão: 0).
 - `size` (int): O número de itens por página (padrão: 20).
 - `sort` (string): O campo para ordenação, seguido pela direção (`asc` ou `desc`). Ex: `titulo,asc`.
- **Resposta de Sucesso (200 OK):**
 - Retorna um objeto de paginação contendo a lista de livros no campo `content`. Cada livro inclui links HATEOAS.

```
{
  "content": [
```

```

{
  "id": 1,
  "titulo": "Arquitetura Limpa",
  "disponivel": true,
  "_links": {
    "self": { "href": "http://localhost:8080/livros/1" }
  }
}
],
"pageable": { ... },
"totalPages": 1,
"totalElements": 3,
...
}

```

-

2. GET /livros/{id}

Busca um livro específico pelo seu ID.

- **Path Parameters:**
 - `id` (Long): O ID único do livro.
- **Resposta de Sucesso (200 OK):**
 - Retorna o objeto completo do livro, incluindo links HATEOAS contextuais.
- **Resposta de Erro:**
 - `404 Not Found`: Se nenhum livro for encontrado com o ID fornecido.

3. POST /livros

Cria um novo livro.

Corpo da Requisição (JSON):

```

{
  "titulo": "Novo Livro",
  "isbn": "978-1234567890",
  "anoPublicacao": 2025,
  "disponivel": true,
  "autorId": 1
}

```

-
- **Resposta de Sucesso (201 Created):**
 - Retorna o objeto do livro recém-criado.
 - Inclui o cabeçalho `Location` com a URL para o novo recurso (ex: `http://localhost:8080/livros/4`).

4. PUT /livros/{id}

Atualiza um livro existente.

- **Path Parameters:**
 - `id` (Long): O ID do livro a ser atualizado.
- **Corpo da Requisição (JSON):**
 - Deve conter todos os campos do livro a serem atualizados.
- **Resposta de Sucesso (200 OK):**
 - Retorna o objeto do livro com os dados atualizados.
- **Resposta de Erro:**

- **404 Not Found:** Se o livro com o ID fornecido não existir.

5. DELETE /livros/{id}

Remove um livro do sistema.

- **Path Parameters:**
 - **id** (Long): O ID do livro a ser removido.
- **Resposta de Sucesso (204 No Content):**
 - Retorna uma resposta vazia, indicando que a operação foi bem-sucedida.
- **Resposta de Erro:**
 - **404 Not Found:** Se o livro com o ID fornecido não existir.