**Nome:** Ismael Felipe Hepp e Daiane Herberts Gonçalves

**Disciplina:** Serviços Web - 6N1

**Desenvolvimento de uma API Rest - Sistema de Livraria**

O sistema da qual a API faz parte é um sistema de uma livraria. Nele, os clientes podem pesquisar por livros e autores. Se o livro estiver em estoque, o preço é apresentado, caso contrário, o cliente pode indicar o interesse pelo livro, recebendo um e-mail avisando quando o livro estiver disponível.

Os funcionários podem acessar o sistema para cadastrar novos livros e autores, atualizar ou deletar os já existentes. Também podem alterar o estoque, quando novos exemplares forem recebidos.

A API Rest desenvolvida faz o controle dos livros da livraria. Nela é possível adicionar, listar, atualizar e deletar registros de livros e pedidos.

A API de livros permite acesso global, enquanto para acessar pedidos é necessária autenticação via gateway.

**Descrição da Tabela ainda não REST**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Path* ou rota do recurso** | **Método HTTP** | **Conteúdo da requisição (parâmetros)** | **Efeito prático em caso de sucesso** |
| \pedidos | GET | nenhum | Retorna lista de todos os pedidos |
| \pedidos | POST | lista de pedidos (ou um único pedido) | Cadastrar um novo pedido no sistema |
| \pedidos | PUT | lista de pedidos | Atualiza a lista de todos os pedidos |
| \pedidos\{id} | GET | nenhum | retorna um pedido com id |
| \pedidos\{id} | PUT | o pedido para o id informado na rota | atualiza pedido com id |
| \pedidos\{id} | DELETE | nenhum | exclui pedido com id informado na rota |

**Mapa de respostas**

|  |  |  |  |
| --- | --- | --- | --- |
| **URI ou rota do recurso** | **Método HTTP** | **Status bem-sucedido** | **Status mal-sucedido** |
| /api/pedidos | GET | 200 | - |
| /api/pedidos/{id} | GET | 200 | 400, 404 |
| /api/pedidos | POST | 201 | 400 |
| /api/pedidos/{id} | PUT | 200 | 400, 404 |
| /api/pedidos/{id} | DELETE | 200 | 400, 404 |

**Recurso**

Abaixo os campos do recurso acessado pela API:

Livro

{

id: Integer,

titulo: String,

quantidade: String,

valor: String,

total: String,

}

**Descrição das rotas e exemplos de requisições**

**Rota /api/pedidos GET**

Uma requisição GET na rota /api/pedidos retorna todos os pedidos cadastrados no sistema.

Exemplo:

Requisição:

curl -X GET \

<http://localhost:3000/api/pedidos> \

-H 'cache-control: no-cache'

Resposta:

HTTP/1.1 200 OK

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 4996

[{

"id": 1,

"titulo": "O Cortiço",

"quantidade": "10",

"valor": "5",

"total": "50"

},

  {

"id": 2,

"titulo": "Era dos Extremos",

"quantidade": "5",

"valor": "7",

"total": "35"

},

  {

"id": 3,

"titulo": "Sejamos todos feministas",

"quantidade": "8",

"valor": "15",

"total": "120"

},

  {

"id": 4,

"titulo": "Quarto de Despejo. Diário de Uma Favelada",

"quantidade": "15",

"valor": "12",

"total": "180"}]

**Rota /api/pedidos/{id} GET**

Uma requisição GET na rota /api/pedidos/{id}, passando um id numérico na url, retorna o pedido cadastrado no sistema que corresponde ao id pesquisado, em caso de sucesso (código 200). Caso o id informado não seja um número, um erro com o código 400 e a mensagem “id deve ser um número” será retornado. Caso o id pesquisado não corresponda a nenhum pedidos cadastrado no sistema, um erro com código 404 e a mensagem “Pedido com id: {id} não encontrado” será retornado.

Exemplo:

Requisição:

curl -X GET \

http://localhost:3000/api/pedidos/3 \

-H 'cache-control: no-cache'

Respostas:

Sucesso (200)

HTTP/1.1 200 OK

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 931

{"id": 3,

"titulo": "Sejamos todos feministas",

"quantidade": "8",

"valor": "15",

"total": "120"}

Falha (400)

HTTP/1.1 400 Bad Request

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 36

{"message":"id deve ser um número"}

Falha (404)

HTTP/1.1 404 Not Found

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 45

{"message":"Pedido com id: 7 não encontrado"}

**Rota /api/pedidos POST**

Uma requisição POST na rota /api/pedidos irá cadastrar um novo pedido no sistema. A requisição deve conter no corpo da mensagem os dados correspondentes a um recurso pedido, exceto o id, com os devidos tipos. Cumprindo este pré-requisito, será retornado o pedido cadastrado e um status de sucesso na criação de um novo recurso (201). Caso qualquer um dos campos exigidos não esteja no formato adequado, será retornado um erro com código 400 e a mensagem “{campo} deve ser {formato}”, onde campo especifica o campo do objeto pedido que não atendeu o pré-requisito e formato o tipo de dado esperado para que o pré-requisito seja atendido.

Exemplo:

Requisição:

curl -X POST \

http://localhost:3000/api/pedidos \

-H 'Content-Type: application/json' \

-H 'cache-control: no-cache' \

-d 'titulo=Teste&isbn=TST-00001111&editora=EDITORA%20TESTE&autor=AUTOR&descricao=DESCRICAO%20TESTE&lancamento=01%2F01%2F2000'

Respostas:

Sucesso (201)

HTTP/1.1 201 Created

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 145

{"id": 5,

"titulo": "O mundo de Sofia",

"quantidade": "3",

"valor": "25",

"total": "75"}

Falha (400)

HTTP/1.1 400 Bad Request

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 40

{"message":"titulo deve ser uma string"}

**Rota /api/pedidos/{id} PUT**

Uma requisição PUT na rota /api/pedidos/{id} irá atualizar os dados de um pedido existente no sistema. A requisição deve conter na url o id do livro a ser atualizado e no corpo da mensagem os dados correspondentes a um recurso pedidos, exceto o id, com os devidos tipos.

Cumprindo estes pré-requisitos, será retornado o pedido atualizado e um status de sucesso (200).

Caso o id informado não seja um número, um erro com o código 400 e a mensagem “id deve ser um número” será retornado. Caso o id pesquisado não corresponda a nenhum pedido cadastrado no sistema, um erro com código 404 e a mensagem “Livro com id: {id} não encontrado” será retornado.

Caso qualquer um dos campos exigidos não esteja no formato adequado, será retornado um erro com código 400 e a mensagem “{campo} deve ser {formato}”, onde campo especifica o campo do objeto pedido que não atendeu o pré-requisito e formato o tipo de dado esperado para que o pré-requisito seja atendido.

Exemplo:

Requisição:

curl -X PUT \

http://localhost:3000/api/pedidos/5 \

Respostas:

Sucesso (200)

HTTP/1.1 200 OK

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 147

{"id": 7,

"titulo": "O mundo de Sofia",

"quantidade": "3",

"valor": "25",

"total": "75"}

Falha (400)

HTTP/1.1 400 Bad Request

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 40

{"message":"titulo deve ser uma string"}

Falha (404)

HTTP/1.1 404 Not Found

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 46

{"message":"Pedido com id: 10 não encontrado"}

**Rota /api/pedidos/{id} DELETE**

Uma requisição DELETE na rota /api/pedidos/{id} irá remover um livro cadastrado do sistema. A requisição deve conter na url o id do pedido a ser removido. Caso o id informado não seja um número, um erro com o código 400 e a mensagem “id deve ser um número” será retornado. Caso o id pesquisado não corresponda a nenhum pedido cadastrado no sistema, um erro com código 404 e a mensagem “Pedido com id: {id} não encontrado” será retornado.

Cumprindo este pré-requisito, será retornado o pedido removido e um status de sucesso (200).

Exemplo:

Requisição:

curl -X DELETE \

<http://localhost:3000/api/pedidos/5> \

-H 'Postman-Token: 5c1df12a-b3a4-4ef9-a5bd-c68159205242' \

-H 'cache-control: no-cache'

Respostas:

Sucesso (200)

HTTP/1.1 200 OK

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 141

{"id":"5","titulo": O mundo de Sofia,"quantidade":"3","valor":"25","total":"TOTAL","descricao":"75"}

Falha (400)

HTTP/1.1 400 Bad Request

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 36

{"message":"id deve ser um número"}

Falha (404)

HTTP/1.1 404 Not Found

X-Powered-By: Express

Content-Type: application/json; charset=utf-8

Content-Length: 47

{"message":"Pedido com id: 100 não encontrado"}