CASOS DE TESTE

São Paulo,

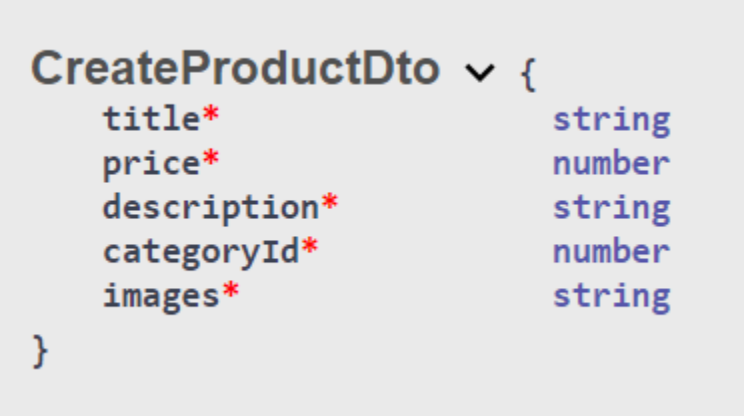
2024

SUMÁRIO

1. CENÁRIO 1 – Criando um novo produto

- Preencher o formulário de produto com “Produto A”, “Descrição do Produto A” e “Preço do Produto A”.
- Clicar no botão salvar.
- Visualizar “Produto A” na lista de produtos

OBSERVAÇÕES: Segundo a documentação Swagger da API, todos os campos são obrigatórios na inclusão:



Então, o usuário precisará informar também o Id categoria e a imagem do produto.

1.1 Caso de teste 1- Cadastro realizado com sucesso

PREMISSA: Cenário onde o cadastro é realizado com sucesso, todas as informações são preenchidas corretamente, salva o produto clicando no botão salvar e o visualiza na lista de produtos.

JSON enviado:

{

"title": "Produto A",

"price": 10,

"description": "Descrição do produto A",

"categoryId": 1,

"images": ["https://placeimg.com/640/480/any"]

}

Response esperado:

{

"title": "Produto A",

"price": 10,

"description": "Descrição do produto A",

"images": [

"[\"https://placeimg.com/640/480/any\"]"

],

"category":

{

"id": 1,

"name": "soy el 9",

"image": "https://i.imgur.com/QkIa5tT.jpeg",

"creationAt": "2024-09-26T14:10:18.000Z",

"updatedAt": "2024-09-26T22:08:20.000Z"

},

"id": 96,

"creationAt": "2024-09-27T13:13:24.000Z",

"updatedAt": "2024-09-27T13:13:24.000Z"

}

- Ao visualizar a lista de produtos, espera-se encontrar:

{

"id": 96,

"title": "Produto A",

```
  "price": 10,
  "description": "Descrição do produto A",
  "images": [
    "[\"https://placeimg.com/640/480/any\"]"
  ],
  "creationAt": "2024-09-27T13:13:24.000Z",
  "updatedAt": "2024-09-27T13:13:24.000Z",
  "category": {
    "id": 1,
    "name": "soy el 9",
    "image": "https://i.imgur.com/QkIa5tT.jpeg",
    "creationAt": "2024-09-26T14:10:18.000Z",
    "updatedAt": "2024-09-26T22:08:20.000Z"
  }
}
```


1.2 Caso de Teste 2 - Falha no cadastro (Falta de descrição do produto)

PREMISSA: Cenário onde no cadastro todas as informações são preenchidas corretamente, porém o campo “description” ficou vazio. Ao clicar no botão, API deve retornar um erro, indicando que não foi possível salvar o cadastro pois “description” não pode ser vazio, já que é um campo obrigatório.

- JSON enviado:


```
{
 "title": "Produto A",
 "price": 10,
 "categoryId": 1,
 "images": ["https://placeimg.com/640/480/any"]
```

}

- Response esperado:

Body:

{

"message":

[

"description should not be empty",

"description must be a string"

],

"error": "Bad Request",

"statusCode": 400

}

1.3 Caso de teste 3: Falha no Cadastro (Falta de descrição e preço do produto)

PREMISSA: Cenário onde no cadastro todas as informações são preenchidas corretamente, porém os campos “description” e “price” ficaram vazios. Ao clicar no botão, API deve retornar um erro, indicando que não foi possível salvar o cadastro pois “description” e “price” não podem ser vazios, já que são campos obrigatórios.

- JSON enviado:

{

"title": "Produto A",

"categoryId": 1,

"images": ["https://placeimg.com/640/480/any"]

}

- Resultado esperado

Body:


```
{
 "message":
 [
          "price should not be empty",
          "price must be a positive number",
          "description should not be empty",
          "description must be a string"
 ],
 "error": "Bad Request",
"statusCode": 400
}
```


1.4 Caso de teste 4 – Falha no Cadastro (Preço enviado como string)

PREMISSA: Cenário onde no cadastro todas as informações são preenchidas corretamente, porém o campo “price” recebe um valor string ao invés de number. Ao clicar no botão, API deve retornar um erro, indicando que não foi possível salvar o cadastro pois “price” não podem ser do tipo string, já que a API espera receber apenas tipo number nesse campo.

- JSON enviado


```
{
     title: "Produto A",
     price: "10",
     description: "Descrição do produto A",
     categoryId: 1,
     images: ["https://placeimg.com/640/480/any"]
   }
```


- Resultado esperado

Assertion Error

expected 10 to equal '10'

2. CENÁRIO 2 – Lendo detalhes do produto

- Clicar no link detalhes do “Produto A”
- Visualizar os detalhes do “Produto A”

PREMISSA GERAL: Espera-se que a API tenha um tratamento para receber valores nos campos, como por exemplo não receber strings parcialmente vazias como “ ”, ou números negativos.

2.1 Caso de teste 1 – Visualização realizada com sucesso

PREMISSA: Nesse cenário, após realizar o cadastro do “Produto A”, ao clicar no botão de visualizar detalhes, deve ser possível encontrar a descrição do produto.

- JSON enviado:

```
{
    title: "Produto A",
    price: 10,
    description: "Descrição do produto A",
    categoryId: 1,
    images: ["https://placeimg.com/640/480/any"]
};
```

- Resultado esperado:

Log Detalhes do produto A: Descrição do produto A

Log Descrição visualizada com sucesso!

2.2 Caso de teste 2 - Produto não possui descrição

PREMISSA: Nesse cenário, cadastramos o campo “description” apenas com um espaço no texto de forma que ele fique assim: “ ”. Após realizar o cadastro do “Produto

A”, espera-se que a API retorne um erro:

- JSON enviado:

```
{
    title: "Produto A",
    price: 10,
    description: " ",
    categoryId: 1,
    images: ["https://placeimg.com/640/480/any"]
  };
```

- Resultado esperado:

Error inserting description;

Result should not have only " "

2.3 Caso de teste 3 – Preço do produto com valor negativo

PREMISSA : Ao realizar o cadastro do “Produto A” com valor negativo, espera-se que API retorne um erro indicando que aceita apenas valores positivos

- JSON Enviado:

```
    {
        title: "Produto A",
        price: -10,
        description: "Descrição do produto A",
        categoryId: 1,
```

images: ["https://placeimg.com/640/480/any"]

}

- Resultado esperado:
  Body:

  {

  "message": ["price must be a positive number"],

  "error": "Bad Request",

  "statusCode": 400

  }

3. CENÁRIO 3 – Atualizando produto existente

- Alterar nome do produto “Produto A” para “Produto B”.
- Clicar no botão de salvar.
- Visualizar “Produto B” na lista de produtos.

3.1 Caso de teste 1 – Atualização realizada com sucesso

PREMISSA: Ao entrar na página de edição do “Produto A”, depois de alterar o nome para “Produto B”, espera-se poder visualizar “Produto B” na lista de produtos.

- JSON enviado:

  body:

  {

  title: "Produto B"

  }

- Resultado esperado:

Log Título alterado e verificado com sucesso: Produto B

4. CENÁRIO 4 – Deletando um produto

- Visualizar lista de produtos.
- Clicar no botão Deletar do “Produto B”.
- Não visualizar “Produto B” na lista de produtos.

4.1 Caso de teste 1 – Produto deletado com sucesso

PREMISSA: Nesse cenário, após visualizar “Produto B”, clicar no botão de deletar do “Produto B”. Assim, espera-se não visualizar o “Produto B” na lista de produtos No método DELETE, é enviado junto da url da API o id do “Produto B”. Dessa forma, um log deve ser enviado como resposta:
Log Produto B não existe mais.