**Plano de Testes - API Restful-Book**

**Introdução**

Este documento descreve o plano de testes para a API do Restful-Booker, um sistema de reservas de hotel. O objetivo é validar a API antes de sua integração com o front-end.

**Cenários de Teste**

2.1 Autenticação

2.1.1 Gerar token de autenticação

Endpoint: POST https://{{baseUrl}}/auth

Request:

{

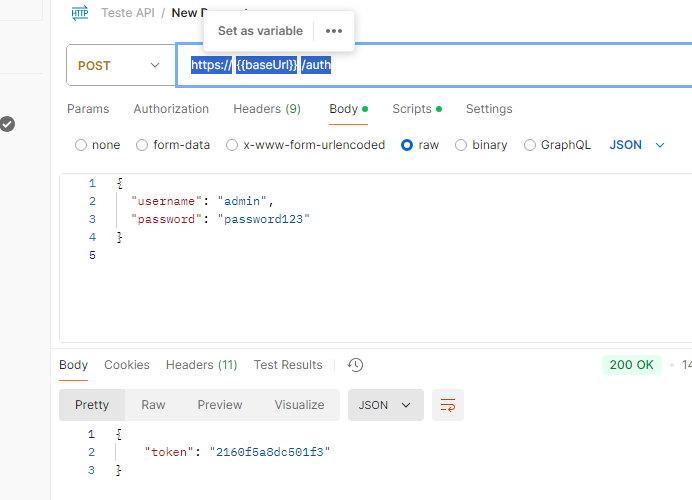
"username": "admin",

"password": "password123"

}

Resultado Esperado: Receber um token de autenticação válido.

Resultado Obtido: Sucesso.



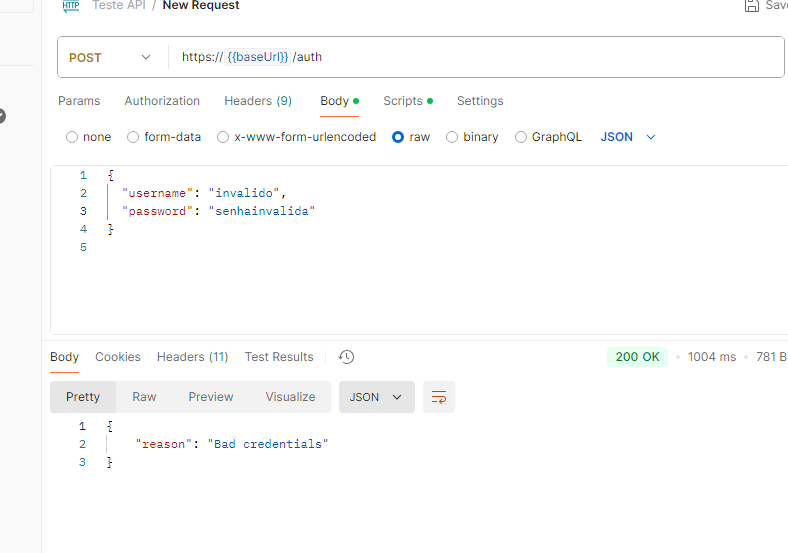
**Tentar gerar token com credenciais inválidas**

**Endpoint**: POST https://{{baseUrl}}/auth

* **Request**: {
* "username": "invalido",
* "password": "invalido"
* }

 **Resultado Esperado**: Receber uma mensagem de erro e status 404.

 **Resultado Obtido**: Recebemos uma mensagem Bad Credentials porem retornou status code 200 então temos um Bug.



### **Gestão de Reservas**

#### Criar uma nova reserva

* **Endpoint**: POST https://{{baseUrl}}/booking
* **Request**:

{

"firstname" : "Jim",

"lastname" : "Brown",

"totalprice" : 111,

"depositpaid" : true,

"bookingdates" : {

"checkin" : "2023-01-01",

"checkout" : "2023-01-10"

},

"additionalneeds" : "Breakfast"

}

 **Resultado Esperado**: A nova reserva é criada e retornada com um ID único.

 **Resultado Obtido**: Sucesso

#### Listar todas as reservas.

* **Endpoint**: GET /booking/
* **Resultado Esperado**: Retornar uma lista de todas as reservas.
* **Resultado Obtido**: Sucesso

#### Atualizar uma reserva existente

* **Endpoint**: PUT /booking/{id}
* **Resultado Esperado**: A reserva é atualizada com as novas informações.
* **Resultado Obtido**: Bad Request erro 403

#### Deletar uma reserva

* **Endpoint**: DELETE /booking/{id}
*  **Resultado Esperado**: A reserva é deletada com sucesso.
* **Resultado Obtido**: Bad Request erro 403.

# **Plano de Testes - Sauce Demo**

## **Introdução**

Este documento descreve o plano de testes para a plataforma de e-commerce Sauce Demo, cobrindo os principais fluxos da aplicação, resultados dos testes.

## **Cenários de Teste**

### **Login com Diferentes Tipos de Usuários**

#### Caso de Teste: Login com usuário padrão

* **Passos**:
  1. Navegar para a página de login.
  2. Inserir nome de usuário válido.
  3. Inserir senha válida.
  4. Clicar no botão de login.
* **Resultado Esperado**: O usuário é redirecionado para a página principal.

#### Caso de Teste: Login com usuário bloqueado

* **Passos**:
  1. Navegar para a página de login.
  2. Inserir nome de usuário bloqueado.
  3. Inserir senha válida.
  4. Clicar no botão de login.
* **Resultado Esperado**: Mensagem de erro indicando que o usuário está bloqueado.

### **Ordenação e Filtragem de Produtos**

#### Caso de Teste: Ordenação por preço (crescente)

* **Passos**:
  1. Navegar para a página de produtos.
  2. Selecionar a opção de ordenação por preço crescente.
* **Resultado Esperado**: Produtos são ordenados do menor para o maior preço.

### **Fluxo Completo de Compra**

#### Caso de Teste: Fluxo de compra

* **Passos**:
  1. Adicionar produtos ao carrinho.
  2. Navegar para o carrinho.
  3. Proceder para o checkout.
  4. Inserir informações de pagamento.
  5. Finalizar compra.
* **Resultado Esperado**: Compra é finalizada com sucesso e uma mensagem de confirmação é exibida.

### **Remoção de Itens do Carrinho**

#### Caso de Teste: Remover item do carrinho

* **Passos**:
  1. Adicionar produtos ao carrinho.
  2. Navegar para o carrinho.
  3. Remover um item do carrinho.
* **Resultado Esperado**: Item é removido do carrinho e o total é atualizado.

### **Navegação Entre Páginas**

#### Caso de Teste: Navegar entre páginas de produtos

* **Passos**:
  1. Navegar para a página principal.
  2. Navegar para diferentes categorias de produtos.
* **Resultado Esperado**: Páginas são carregadas corretamente e sem erros.

### **Logout**

#### Caso de Teste: Logout

* **Passos**:
  1. Estar logado na plataforma.
  2. Clicar no botão de logout.
* **Resultado Esperado**: Usuário é redirecionado para a página de login.