



ServeRest
RELATÓRIO DE TESTES

LogicalForest 2

Felipe Ripplinger Dupont

1.Introdução

Bem vindo ao relatório de testes da API ServeRest, aqui irei tratar dos resultados encontrados na execução do conteúdo do [plano de testes](#).

2.Análise de resultados

Todos os 42 casos de teste apresentados no plano de testes foram automatizados com sucesso, com suas etapas, se totalizam em 53 na contagem do framework Cypress.

Os testes podem ser realizados tanto em ambiente de produção quanto em ambiente local. Durante o desenvolvimento me comprometi que a execução dos testes regressivos não afetasse o ambiente excessivamente, tudo que fora criado pelos testes também deveria ser excluído pelos testes, assim criando um ciclo que não deixaria muitos resíduos, entretanto como citei isso foi nos casos dos 40 testes regressivos, no caso dos 2 testes de fluxo, seus usuários ainda se mantêm após a execução para permanecer fiel ao seu propósito.

3.Cobertura de testes

Como almejado no plano de testes, o path coverage foi de 100%, tratando todas as funcionalidades e suas possíveis falhas documentadas, com exceção duas que gostaria de citar, a primeira é a “função” POST do método PUT na rota usuários, já que ela apenas simula uma função que já está sendo testada e o método para validar o erro de token, foi apenas o token ausente, não tratando das demais.

3. Erros encontrados

BUG-01
Local afetado: Rotas que exijam uma quantidade do body da requisição.
Descrição: Caso a quantidade a ser enviada seja um número muito grande (maior que 9007199254740991) ou não válido, a api considera como um número não seguro e não envia ele na requisição, então o erro retornado é " não contém 1 valor obrigatório".

Sugestão de correção:

Criar uma mensagem de erro específica para a situação, como “O valor da quantidade não é um número válido ”

BUG-02**Local afetado:**

[POST /login].

Descrição:

O erro de email e/ou senha inválidos, na documentação está como se retornasse um código 400, quando na verdade ele retorna um código 401.

Sugestão de correção:

Ajustar o código de erro na documentação ou na API para que fiquem idênticos.

BUG-03**Local afetado:**

[GET /carrinhos] e [GET /carrinhos/{_id}].

Descrição:

Os endpoints afetados estão com o body resposta incorreta na documentação, possuindo um par de colchetes extra dentro da chave produtos.

Sugestão de correção:

Ajustar o body da resposta na documentação ou na API para que fiquem idênticos.

BUG-04**Local afetado:**

[GET /produtos].

Descrição:

O endpoint afetado está com o body resposta incorreta na documentação, possuindo a chave usuários ao invés de produtos.

Sugestão de correção:

Ajustar o body da resposta na documentação para que fique idêntico ao da API.

BUG-05**Local afetado:**

Todas as rotas que exijam um body com algum valor.

Descrição:

Ao enviar um body com valores sem aspas, o código de erro retornado é um 500 que não está documentado, o erro também acontece se o body estiver de alguma outra maneira errado. Esse problema não pode ser visto através do site da API apenas por requisições

de terceiros.

Sugestão de correção:

Desenvolver mais este código de erro para que ele se adapte melhor a situação e registrar ele na documentação.

4.Sugestões gerais

- Adicionar os códigos de erro 5xx na documentação.
- Um método PUT para a rota carrinhos, uma função básica, entretanto essencial para qualquer sistema que use carrinhos.
- A implementação do código 404 seria uma boa ideia para rotas não existentes, atualmente elas retornam o erro 405, que segundo o RFC 7231, indica que o método recebido na requisição é conhecido pelo servidor de origem, mas não é suportado pelo recurso de destino, algo que não se encaixa muito bem.