

Postman - Report de Bugs - Squad Ruptura

POST /user – Retorno incorreto de status code e falha na validação de duplicidade de usuário

1- Descrição detalhada

Durante os testes da rota POST/user da API da Petstore, foi identificado um comportamento inconsistente no retorno de *status codes*. Quando um novo usuário é cadastrado, a API retorna o status code 200 OK, em vez do esperado 201 Created. Além disso, ao tentar cadastrar um usuário já existente (mesmo ID ou mesmo username), a API retorna 500 Internal Server Error, quando o ideal seria um erro controlado como o 409 Conflict ou uma mensagem explicativa.

Também foi possível cadastrar usuários com o mesmo username, modificando-se apenas o ID, o que compromete a unicidade do identificador principal de login.

2- Passo a passo

- Acesse a rota: <https://petstore3.swagger.io/api/v3/user>
- Execute o cadastro de um novo usuário com dados válidos.
- Realize os seguintes testes subsequentes:

- c.1.** Envie novamente o mesmo payload (mesmo ID e username) → **Retorno: 500**
- c.2.** Altere apenas o username, mantendo o mesmo ID → **Retorno: 500**
- c.3.** Altere apenas o ID, mantendo o mesmo username → **Retorno: 200 e cadastro realizado**

3- Resultado(s) esperado(s)

- O sistema deve retornar o status code 201 ao realizar a criação de um novo usuário com sucesso.
- O sistema deve retornar 409 Conflict ou erro similar ao tentar cadastrar um usuário com ID ou username já existente.
- O sistema deve apresentar uma mensagem clara explicando o motivo da falha no cadastro.
- O sistema não deve permitir múltiplos cadastros com o mesmo username.

4- Resultado Atual

- O sistema retorna status 200 OK ao criar um novo usuário.
- Ao tentar cadastrar um usuário já existente, a API retorna status 500 Internal Server Error sem mensagem descritiva.
- O sistema permite o cadastro de múltiplos usuários com o mesmo username, desde que o ID seja diferente.

5 - Gravidade

- ☒ Alta - O defeito impacta diretamente a integridade dos dados e a experiência do usuário, podendo apresentar um risco de falha de autenticação.
- ☐ Média
- ☐ Baixa

6 - Prioridade

- ☒ Imediata - O problema precisa ser corrigido com urgência, pois compromete a lógica de negócio e a confiabilidade do sistema.
- ☐ Alta
- ☐ Média
- ☐ Baixa

7 - Ambiente onde ocorre

- Ambiente: Produção (Swagger Petstore - API pública)
- URL: [Swagger UI](#)
- Rota testada: POST/user
- Navegador: Google Chrome
- Ferramentas: Swagger UI, Postman

8 - Responsável

Equipe de Desenvolvimento

9 - Anexos



10 - Relação com outros defeitos ou requisitos

Login - a duplicidade de usernames pode impactar a funcionalidade de autenticação.

11 - Notas Adicionais:

- Verificar se existe política de unicidade definida para campos como username e ID.
- Recomenda-se validação prévia e tratamento adequado para retorno de erros, evitando o uso de status 500 em casos previstos. <https://luisms.atlassian.net/browse/SR-19>

Post/pet/{petId}/uploadImage: Erro 415 ao fazer upload de imagem para pet [🔗](#)

1. Descrição detalhada

Durante os testes da rota POST /pet/{petId}/uploadImage, foi identificado que o sistema retorna “415 Unsupported Media Type” mesmo quando os dados são enviados corretamente no formato `form-data`.

2. Passo a passo para reproduzir

1. Acesse a rota no Postman: <https://petstore.swagger.io/v2/pet/100/uploadImage>
2. Configure o Body como form-data com os seguintes campos:
 - additionalMetadata: Text → 1
 - file: File → th.jpeg (imagem válida)
3. Clique em "Send"

3. Resultado(s) esperado(s)

- A API deve aceitar o envio multipart e retornar status 200 OK ou similar.

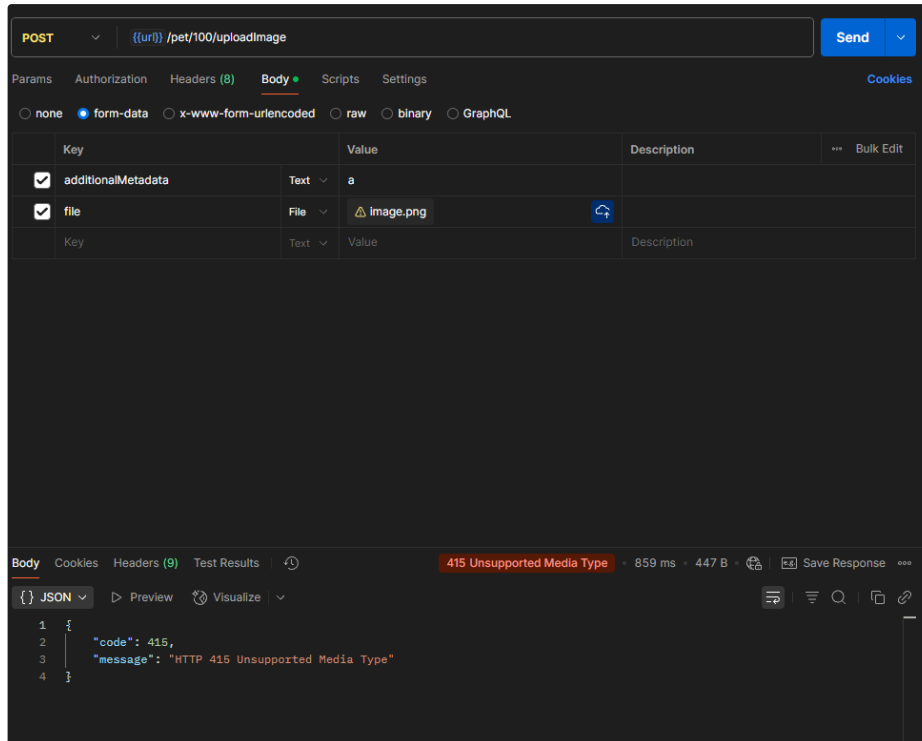
4. Resultado atual

Gera um erro “415 Unsupported Media Type”

Conforme a documentação do Swagger disponível no site: [Media Types / Swagger Docs](#) → o Swagger suporta o envio de arquivos de imagem no formato .png, no entanto, nessa requisição foi encontrado o erro 415.

5. Anexos

6.



DELETE/order/{orderId} - Posso colocar um ID acima de 1000, o que não deveria ser possível. [🔗](#)

1- Descrição detalhada:

De acordo com o Swagger da API Petstore , ao acessar a rota “Store” e ir no request “DELETE” para excluir um pedido por ID, ao clicar para ler as informações do request e então começar a testar, nos informa que: “Para uma resposta válida, tente IDs inteiros com valor < 1000. Qualquer valor acima de 1000 ou não inteiros gerará erros de API.”

Enquanto eu estava testando, observei essa informação e tentei colocar um valor acima de 1000, resultado: Não gerou erros de API, apenas retornou código 200 OK

2- Passo a passo:

- No postman:
 - crie uma collection
 - crie as pastas nomeadas com as rotas da api Petstore e seus respectivos requests
 - Pet
 - Store
 - User
 - Dentro da pasta Store, após criar os outros requests, criar o request “DELETE”
 - Na url do request, adicionar a rota: <https://petstore3.swagger.io/api/v3/store/order/>
- Na url, após o “order ”, adicionar um valor inteiro (IDs) maior que 1000.
- Ao executar o teste, irá receber a response 200 OK.

3- Resultado(s) esperado(s):

- Ao inserir qualquer valor inteiro como ID maior que 1000, gerar erros na API.

4- Resultado Atual:

- Ao inserir um valor inteiro como ID maior que 1000, está retornando como resposta: 200 OK

5 - Gravidade:

- ☐ Alta
- ☒ Média - Caso a funcionalidade não seja central para o fluxo de negócios e o impacto for apenas na documentação e no entendimento do usuário.
- ☐ Baixa

6 - Prioridade:

- ☐ Imediata
- ☐ Alta
- ☒ Média - Caso o impacto seja limitado e não afete diretamente a operação principal.
- ☐ Baixa

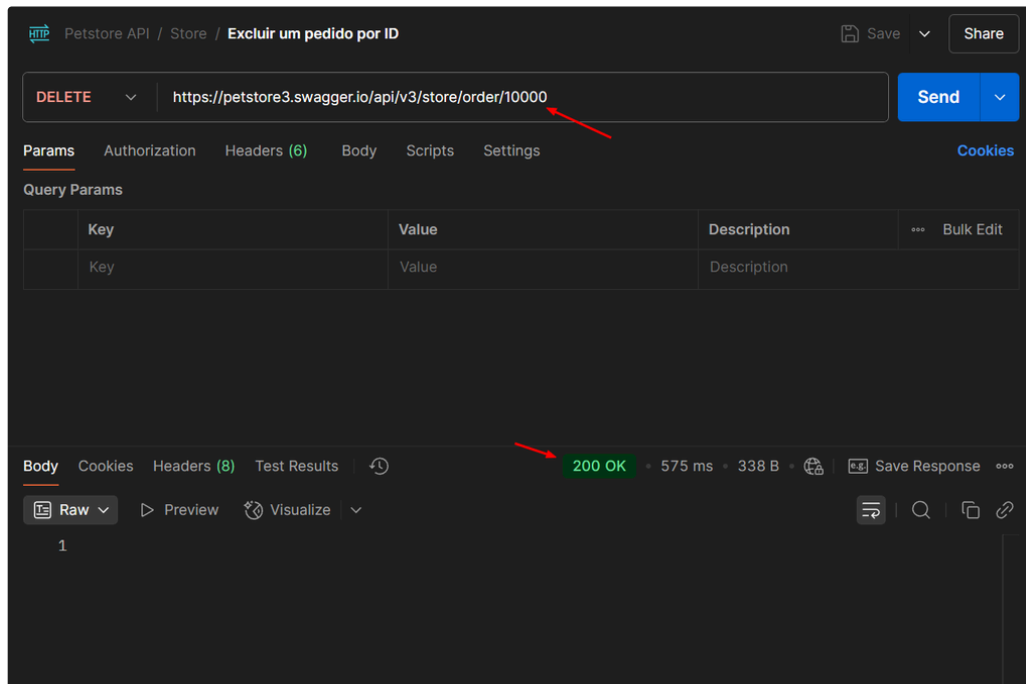
7 - Ambiente onde ocorre:

- Ambiente: Produção (Swagger Petstore - API pública)
- URL: <https://petstore3.swagger.io/>
- Rota testada: Store
- Navegador: Google Chrome
- Ferramentas: Postman

8 - Responsável:

- Equipe de Desenvolvimento

9 - Anexos:



10 - Relação com outros defeitos ou requisitos:

- Não foi observado nenhuma relação com outros defeitos ou requisitos.

11 - Notas Adicionais:

- Recomenda-se revisar outros endpoints na rota 'Store' para verificar se o problema de validação de IDs ou respostas inconsistentes está presente em mais casos.

- Pode ser útil verificar os padrões de resposta e validação na API, especialmente para parâmetros numéricos, para garantir consistência em todos os endpoints.
 - Este problema pode indicar uma desconexão geral entre a documentação Swagger e a implementação real da API.
-

GET/user/login - Usuário consegue fazer login utilizando nome de usuário e/ou senha inválidos.

1- Descrição detalhada:

De acordo com o Swagger da API Petstore , ao acessar a rota “User” e ir no request “GET” para registrar o usuário no sistema (fazer login), ao clicar para ler as informações do request e então começar a testar, observo que nas respostas que deveriam retornar, uma delas tem o código 400 e sua descrição é: “Nome de usuário/senha fornecidos inválidos”.

Enquanto estava testando esse request, decidi testar essa informação. Para testar isso, resolvi tentar fazer um login sem fornecer nome de usuário e senha, o que deveria ser considerado inválido. Ao testar, observei que retorna 200 OK, ao invés de um código 400.

2- Passo a passo:

1. No postman:
 - a. crie uma collection
 - b. crie as pastas nomeadas com as rotas da api Petstore e dentro de cada rota, criar seus respectivos requests
 - i. Pet
 - ii. Store
 - iii. User
 - c. Dentro da pasta User, um dos requests para criar é o “GET” para o usuário realizar login.
 - i. Na url desse request, adicionar a rota: <https://petstore3.swagger.io/api/v3/user/login>
2. Dentro do request “GET”, clicar nos “Params”, ao acessar, vai aparecer “Query Params” e campos para adicionar uma key, seu value e uma description
3. No campo “Key”, clicar em cima e adicionar o nome “username”, no campo “Value”, clicar em cima e digitar o nome do usuário.
4. Embaixo do parâmetro criado acima, clicar no campo “Key” e adicionar o nome “password”, no campo “Value”, cliço em cima e adiciono uma senha.
5. Após isso, cliço no quadrado do primeiro campo que está escrito “Key” , “Value” e “Description”, para realizar o teste com os parâmetros selecionados e ver a resposta 200 OK
6. Então, cliço novamente neste quadrado do primeiro campo, para desselecionar os parâmetros e realizar o teste sem um nome de usuário e uma senha válida. E observo retornar a resposta 200 OK

3- Resultado(s) esperado(s):

- Ao tentar realizar o login com nome de usuário ou senha inválidos, não deveria ser permitido e deveria retornar um código 400.

4- Resultado Atual:

- Ao tentar realizar login com usuário ou senha inválidos, é permitido e retorna um código 200 OK.

5 - Gravidade:

- ☒ Alta - A funcionalidade de login é central para a segurança e operação da maioria dos sistemas.
- ☐ Média
- ☐ Baixa

6 - Prioridade:

- ☒ Imediata - A falta de validação no login pode expor o sistema a vulnerabilidades críticas.

- ☐ Alta
- ☐ Média
- ☐ Baixa

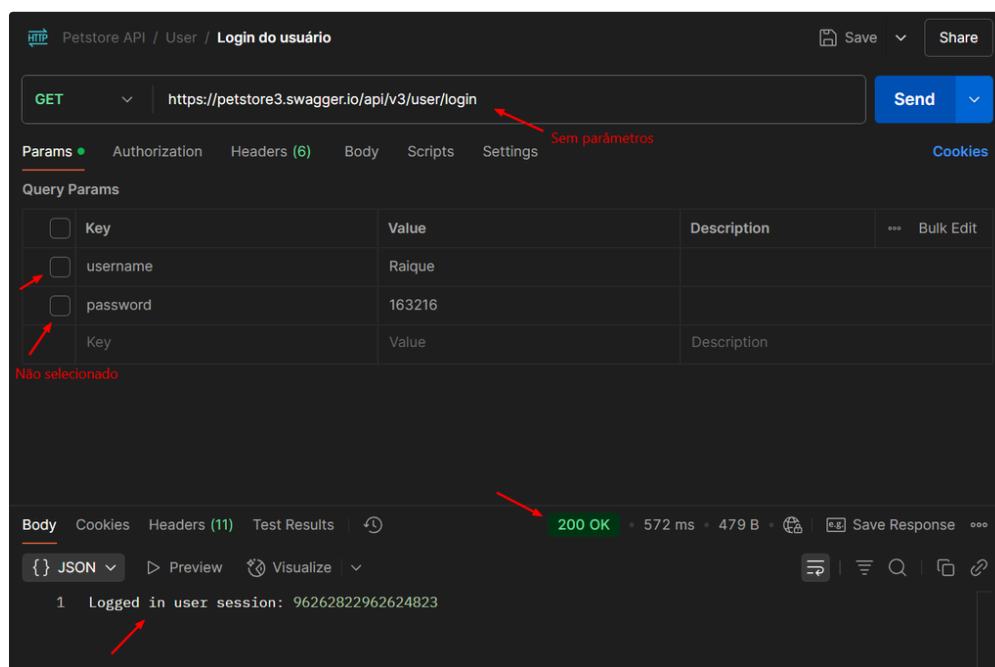
7 - Ambiente onde ocorre:

- Ambiente: Produção (Swagger Petstore - API pública)
 - URL: <https://petstore3.swagger.io/>
 - Rota testada: GET/user/login
 - Navegador: Google Chrome
 - Ferramentas: Postman

8 - Responsável:

- Equipe de Desenvolvimento

9 - Anexos:



10 - Relação com outros defeitos ou requisitos:

- Não foi observado nenhuma relação com outros defeitos ou requisitos.

11 - Notas Adicionais:

Embora não tenha sido observado relação com outros defeitos ou requisitos, podem existir algumas relações, como:

- Se o mesmo padrão de validação inadequada for observado em outros endpoints, o problema não é isolado e pode indicar uma falha de design ou implementação generalizada na API.
- O problema relatado também reflete uma discrepância entre a documentação (Swagger) e o comportamento real da API.
 - Se este problema ocorre em um endpoint tão crítico como o login, há uma chance de que a documentação de outros endpoints também esteja inconsistente.
 - Isso pode exigir uma revisão geral da documentação Swagger para garantir que esteja alinhada ao comportamento real da API.
- Uma validação inadequada no login pode comprometer a segurança e a integridade de operações subsequentes que dependem de autenticação.

PUT /pet - Permite método POST [↗](#)

1. Descrição detalhada [↗](#)

Durante os testes exploratórios da rota PUT /pet, foi verificado que o endpoint também aceita o método POST, retornando 200 OK, mesmo que o método oficial esperado seja PUT para atualização de um pet existente.

2. Passo a passo para reproduzir [↗](#)

- Definir o corpo JSON com os dados completos de um pet existente (com id, name, etc.)
- Enviar a requisição para {{url}}/pet
- Alterar o método para ****POST****
- Observar a resposta da API

3. Resultado esperado [↗](#)

- A API deveria rejeitar métodos não suportados na rota, retornando por exemplo:
 - 405 Method Not Allowed

4. Resultado atual

- A API responde com 200 OK
- O pet é possivelmente criado ou sobrescrito, sem validar o verbo HTTP

5. Gravidade [↗](#)

Alta

6. Prioridade [↗](#)

Alta

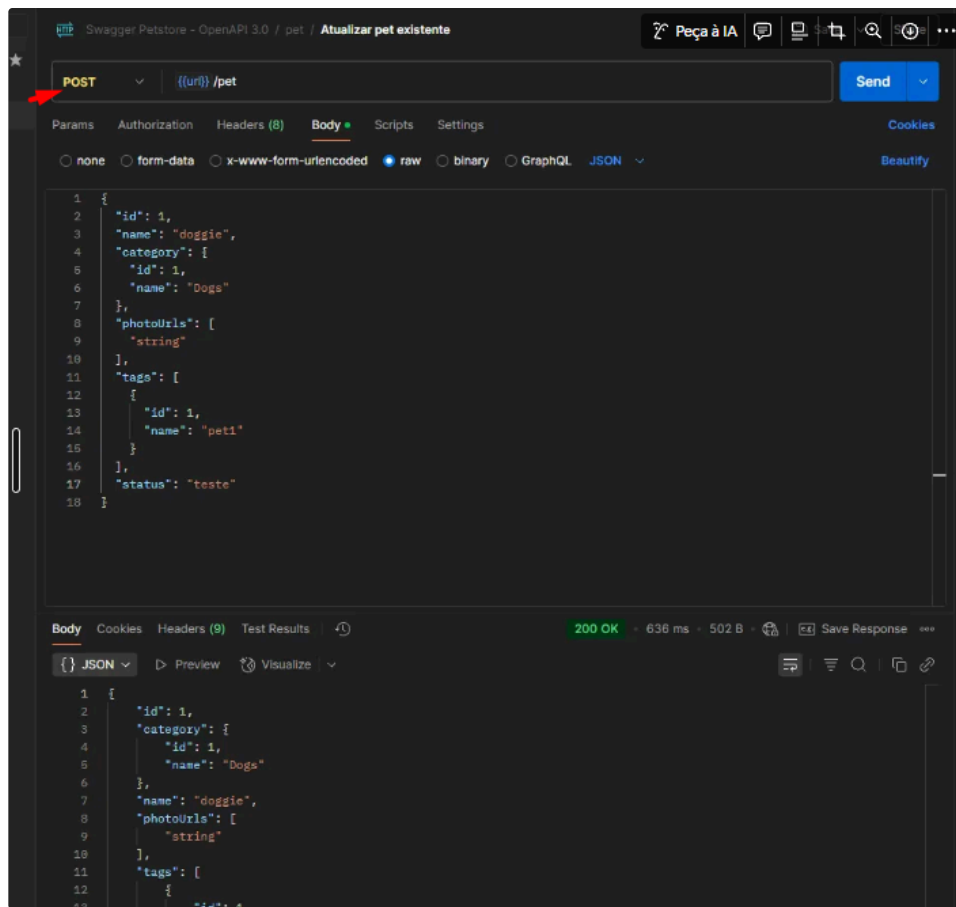
7. Ambiente [↗](#)

- Ambiente: Produção (Swagger Petstore pública)
- URL: <https://petstore.swagger.io/v2/pet>
- Ferramenta usada: Postman

8. Responsável [↗](#)

Equipe de desenvolvimento

9. Anexos [↗](#)



PUT/pet - permite atualizar pet com campos obrigatórios vazios [🔗](#)

1. Descrição detalhada [🔗](#)

Durante o teste da rota PUT /pet, foi identificado que o sistema permite que um pet existente seja atualizado com valores vazios nos seguintes campos obrigatórios:

- name do pet
- category.name
- tag.name
- status

Mesmo com todos esses campos vazios, a API responde com ****200 OK****, indicando sucesso na operação — sem validação mínima dos dados enviados.

2. Passo a passo para reproduzir [🔗](#)

1. Enviar requisição PUT para {{url}}/pet
2. Corpo da requisição JSON deixar vazio os campos:

- name do pet
- category.name
- tag.name
- status

1. Verificar a resposta da API

3. Resultado esperado [🔗](#)

- A API deveria retornar:
 - 400 invalid ID supplied ou
 - 422 Validation exception

4. Resultado atual [↗](#)

- A API retorna 200 OK
- Atualiza o recurso com valores vazios
- Não valida a integridade dos dados

5. Gravidade [↗](#)

Alta

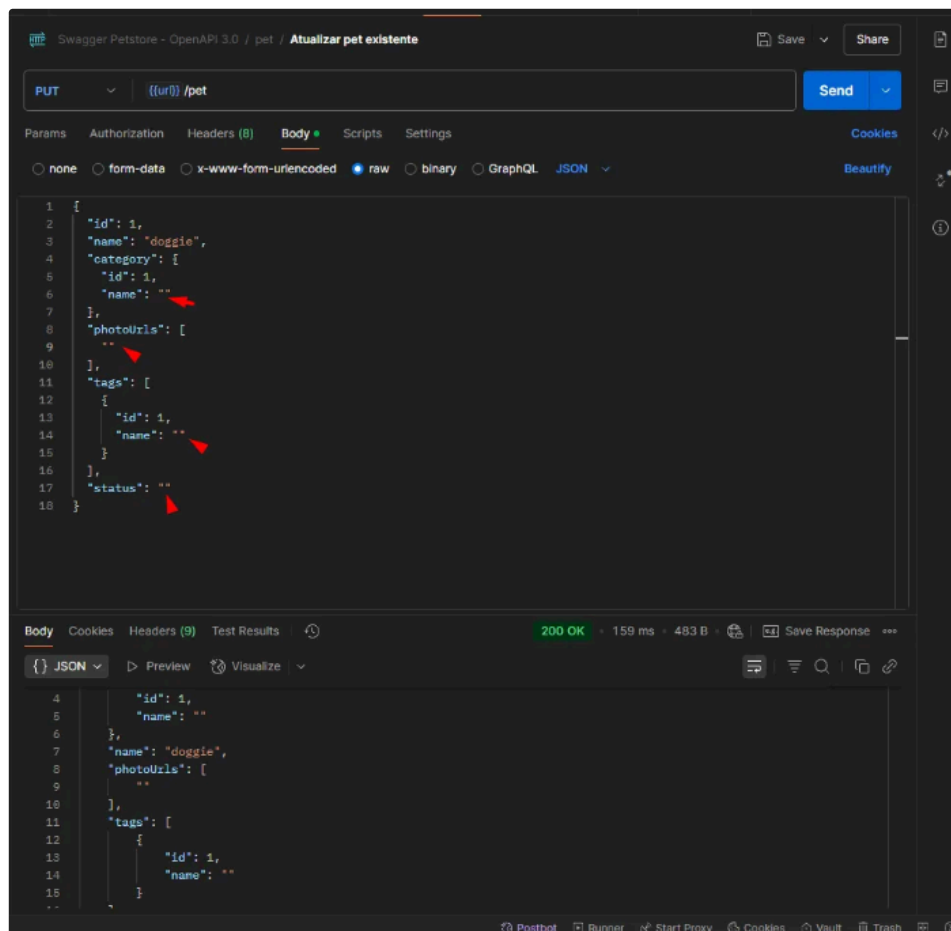
6. Prioridade [↗](#)

Imediata

7. Ambiente onde ocorre [↗](#)

- Ambiente: Produção (Swagger Petstore pública)
- Ferramenta: Postman
- Rota: PUT /pet
- URL: <https://petstore.swagger.io/v2/pet>

8. Evidência (anexo) [↗](#)



DELETE/store permite exclusão com ID inválido (acima de 1000)

1. Descrição detalhada:

Durante o teste da rota DELETE /store/order/{orderId}, foi identificado que a API permite excluir pedidos com orderId superior a 1000, mesmo que a documentação informe que valores maiores que 1000 devem gerar erros.

2. Passo a passo para reprodução:

- a. Abra o Postman ou Swagger UI
- b. Execute a requisição: DELETE /store/order/1001
- c. Observe o status de resposta e o corpo retornado

3. Resultado Esperado:

A API deveria retornar 400 Invalid ID supplied já que o ID informado (1001) excede o limite aceito documentado.

4. Resultado Atual:

A API retorna 200 OK, mesmo com orderId = 1001, sem validar corretamente o valor fornecido.

5. Gravidade:

Alta

6. Prioridade:

Alta

7. Ambiente onde ocorre:

- Ambiente: Produção (Swagger Petstore - API pública)
- URL: <https://petstore3.swagger.io>
- Rota testada: DELETE /store/order/{orderId}
- Navegador: Google Chrome

8. Responsável:

Equipe de Desenvolvimento

9. Anexos:

Swagger Petstore - OpenAPI 3.0 / store / Excluir pedido por ID

Save Share

DELETE

{{uri}} /store/order1001

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body

Cookies

Headers (8)

Test Results

200 OK

144 ms

338 B

Save Response

Raw

Preview

Visualize

1