

Justificativas para falhas nos testes

Ricardo Ribeiro Martins

@ResetLevel2 @TestesAPI @SmokeTests

C0001 - [GET] Listar IDs de reservas utilizando o filtro 'checkin'

Dado que preencho parâmetro **checkin** com o valor contido no último booking cadastrado

Quando envio um GET Request para "https://treinamento-api.herokuapp.com/**booking**"

Então retorna lista contendo ao menos o objeto "**bookings : id**" do último booking cadastrado

Criticidade: MÉDIA

Prioridade: 3

@ResetLevel2 @TestesAPI @SmokeTests

C0002 - [GET] Listar IDs de reservas utilizando os filtros 'checkin' e 'checkout'

Dado que preencho parâmetro **checkin** com o valor contido no último booking cadastrado

E preencho parâmetro **checkout** com o valor contido no último booking cadastrado

Quando envio um GET Request para "https://treinamento-api.herokuapp.com/**booking**"

Então retorna lista contendo ao menos o objeto "**bookings : id**" do último booking cadastrado

Criticidade: MÉDIA

Prioridade: 3

@ResetLevel2 @TestesAPI @SmokeTests

C0003 – [GET] Listar IDs de reservas utilizando os filtros 'firstname', 'checkin' e 'checkout'

Dado que preencho parâmetro **firstname** com o valor contido no último booking cadastrado

E preencho parâmetro **checkout** com o valor contido no último booking cadastrado

E preencho parâmetro **checkout** com o valor contido no último booking cadastrado

Quando envio um GET Request para "https://treinamento-api.herokuapp.com/**booking**"

Então retorna lista contendo ao menos o objeto "**bookings : id**" do último booking cadastrado

Criticidade: MÉDIA

Prioridade: 2

@ResetLevel2 @TestesAPI @AcceptanceTests

C0004 - [GET] Visualizar erro de servidor 500 quando enviar filtro mal formatado

Dado que preencho um parâmetro **inválido** com qualquer valor para ele

Quando envio um GET Request para "https://treinamento-api.herokuapp.com/**booking**"

Então retorna Status Code <500> sem resultados

Criticidade: Média

Prioridade: 2

@ResetLevel2 @TestesAPI @AcceptanceTests @SecurityTests

C0005 – [PUT] Tentar alterar uma reserva quando o token não for enviado

Dado que preencho *header* com *Content-Type = application/json*

E *header* com *Accept = application/json*

E *body* com *payload* válido e completo

Quando envio um PUT Request para "https://treinamento-api.herokuapp.com/**booking/:id_existente**"

Então retorna Status Code <401> e a alteração do booking não é efetuada

Criticidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @AcceptanceTests

C0006 - [PUT] Tentar alterar uma reserva que não existe

Dado que preencho *header* com *Content-Type = application/json*

E *header* com *Accept = application/json*

E *header* *Cookie* com informação válida

E preencho o *body* com *payload* válido completo

Quando envio um PUT Request para "https://treinamento-api.herokuapp.com/**booking/:id_inexistente**"

Então retorna Status Code <404> e nenhuma alteração é realizada

Criticidade: BAIXA

Prioridade: 3

@ResetLevel2 @TestesAPI @SmokeTests @SecurityTests

C0007 - [PUT] Alterar uma reserva usando o Basic auth informado na documentação de Booking - UpdateBooking .Header

Dado que preencho header com *Content-Type = application/json*

E preencho header com *Accept = application/json*

E preencho header com *Authorisation = Basic YWRtaW46cGFzc3dvcmQxMjM=J*

E preencho *body* com payload válido completo

Quando envio PUT Request para "https://treinamento
api.herokuapp.com/**booking/:id_existente**"

Então retorna Status Code <200> e realiza a alteração do booking especificado

Criticidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @SmokeTests @SecurityTests

C0008 – [POST] Tentar criar uma reserva sem estar logado

Dado que preencho header com *Content-Type = application/json*

E header com *Accept = application/json*

E não forneço um header *Cookie*

E preencho o *body* com payload válido completo

Quando envio um POST Request para
"https://treinamento-api.herokuapp.com/**booking/:id_inexistente**"

Então retorna Status Code <401> e nenhum booking é criado

Criticidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @AcceptanceTests @ContractTests @SecurityTests

C0009 - [POST] Criar uma reserva enviando mais parâmetros no payload da reserva

Dado que preencho header com *Content-Type = application/json*

E preencho header *Accept = application/json*

E preencho header *Cookie* com token válido de autenticação

E preencho *body* com payload completo com um parâmetro adicional
("language" : "portuguese")

Quando envio POST Request para "https://treinamento
api.herokuapp.com/**booking/**"

Então retorna Status Code <400> e não efetua a criação do booking

Criticidade: MÉDIA

Prioridade: 2

@ResetLevel2 @TestesAPI @SmokeTests @SecurityTests

C0010 - [DELETE] Excluir um reserva com sucesso

Dado que preencho header com *Content-Type = application/json*

E preencho header *Cookie* com token válido de autenticação

Quando envio DELETE Request para

"https://treinamento api.herokuapp.com/**booking/:id_existente**"

Então retorna Status Code <204> e deleta o booking requerido

Criticidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @AcceptanceTests

C0011 - [DELETE] Tentar excluir um reserva que não existe

Dado que preencho header com *Content-Type = application/json*

E preencho header *Cookie* com token válido de autenticação

Quando envio DELETE Request para

"https://treinamento api.herokuapp.com/**booking/:id_inexistente**"

Então retorna Status Code <404> e informa sobre ausência do booking

Criticidade: BAIXA

Prioridade: 3

@ResetLevel2 @TestesAPI @AcceptanceTests @SmokeTests @SecurityTests

C0012 - [POST] Validar a criação de mais de uma reserva em sequência

Dado que preencho header com *Content-Type = application/json*

E header com *Accept = application/json*

E forneço um header *Cookie* válido

E preencho o *body* com payload válido completo

Quando envio três POST Request consecutivos para
"https://treinamento-api.herokuapp.com/**booking/**"

Então retorna Status Code <429> após o terceiro e impede ações adicionais

Criticidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @SmokeTests

B0001 – Pesquisa filtrada por parâmetro **checkin** não retorna informação pretendida

Método GET não está retornando resultado esperado com filtro **checkin**

Passos para reproduzir:

- 1- Preencher parâmetro **checkin** com o respectivo valor do último booking
- 2- Realizar GET Request para "https://treinamento-api.herokuapp.com/**booking**"
- 3 – Solicitação é executada com sucesso, mas retorna lista que não contém o último booking

Resultado esperado:

A lista retornada deve conter ao menos o objeto "**booking : id**" do último booking, mas também outros que tenham data de **checkin** igual ou posterior à parametrizada

Observações:

- 1- Ao realizar pesquisa colocando ao menos **o dia anterior** a data de **checkin** do booking pretendido, ele **retorna** o mesmo com sucesso, o que demonstra que este parâmetro não é inclusivo
- 2- A lista retornada contém todos os bookings a partir do **dia seguinte** à data de **checkin** parametrizada

Evidência:

Todos os **ids** retornados incluem uma data de **checkin** ao menos um dia posterior ao valor atribuído ao parâmetro na pesquisa, mas não aquelas **a partir** daquele dia.

```

1 expectation failed.
JSON path bookingid doesn't match.
Expected: a collection containing <703>
Actual: [624, 638, 704, 595, 307, 409, 388, 577, 582, 594,
644, 581, 681, 306, 217, 579, 591, 593, 178, 118, 592, 604, 576,
567, 357, 221, 218, 285, 138, 384, 449, 605, 18, 613, 359, 617,
609, 58, 369, 125, 441, 590, 616, 349, 215, 123, 75, 540, 26,
31, 614, 56, 737, 390, 216, 417, 135, 424, 598, 74, 304, 627,
17, 8, 574, 134, 303, 393, 649, 575, 701, 129, 620, 601, 533,
301, 365, 170, 383, 682, 416, 43, 155, 448, 49, 6, 126, 32, 402,
658, 35, 45, 318, 308, 546, 120, 367, 321, 59, 151, 223, 600,
621, 214, 34, 450, 410, 55, 44, 78, 659, 147, 146, 446, 41, 392,
165, 171, 635, 618, 738, 10, 447, 607, 578, 88, 419, 685, 641,
143, 132, 127, 154, 550, 549, 623, 647, 420, 152, 316, 30, 499,
315, 385, 53, 57, 25, 368, 121, 634, 222, 403, 119, 153, 423,
637, 596, 327, 626, 89, 606, 295, 386, 534, 425, 164, 622, 608,
33, 619, 639, 305, 130, 422, 15, 320, 122, 169, 537, 389, 133,
39, 610, 611, 317, 573, 720, 394, 136, 572, 27]

```

Cenário relacionado: C0001.

Severidade: MÉDIA

Prioridade: 2

@ResetLevel2 @TestesAPI @SmokeTests

B0002 – Pesquisa filtrada por parâmetro **checkin** e **checkout** não retorna informação pretendida

Método GET não está retornando resultado esperado com filtro **checkin** e **checkout** juntos

Passos para reproduzir:

- 1 – Preencher parâmetros **checkin** e **checkout** com os respectivos valores do último booking
- 2- Realizar GET Request para "https://treinamento-api.herokuapp.com/**booking**"
- 3 – Solicitação é executada com sucesso, mas retorna lista que não contém o último booking

Resultado esperado:

A lista deve retornar contendo ao menos o objeto "**booking : id**" do último booking, mas também outros que estejam dentro do intervalo entre **checkin** e **checkout**.

Observações:

- 1- Ao realizar pesquisa colocando ao menos o **dia anterior** a data de **checkin** do booking pretendido, ele **retorna** o mesmo com sucesso, o que demonstra que este parâmetro não é inclusivo, e o parâmetro **checkout**, sim
- 2- A lista retornada contém todos os bookings a partir do **dia seguinte** à data de **checkin** parametrizada até a data exata de **checkout** parametrizada

Evidência:

Todos os **ids** retornados incluem uma data de **checkin** ao menos um dia posterior ao valor atribuído ao parâmetro na pesquisa, mas não aquelas **a partir** daquele dia.

```
1 expectation failed.  
JSON path bookingid doesn't match.  
Expected: a collection containing <703>  
Actual: [449, 441, 75, 31, 74, 448, 32, 214, 34, 450, 78, 446,  
447, 30, 89, 33]
```

Cenário relacionado: C0002.

Severidade: MÉDIA

Prioridade: 2

@ResetLevel2 @TestesAPI @SmokeTests

B0003 – Pesquisa filtrada por parâmetros **firstname**, **checkin** e **checkout** não retorna informação pretendida

Método GET não está retornando lista por filtro **checkin** e **checkin** juntos

Passos para reproduzir:

- 1 – Preencher parâmetros **firstname**, **checkin** e **checkin** com os exatos valores do último booking
- 2- Realizar GET Request para o path "https://treinamento-api.herokuapp.com/**booking**"
- 3 – Solicitação é executada com sucesso, mas retorna lista não contendo o último booking

Resultado esperado:

A lista deve retornar contendo ao menos o objeto "**booking : id**" do último booking, mas também outros que estejam dentro do intervalo de **checkin** e **checkout** e tenham o mesmo **firstname**.

Observações:

- 1- Ao realizar pesquisa colocando **o dia anterior** a data de **checkin** do booking de interesse, ele **retorna** a mesma com sucesso, o que demonstra que este parâmetro não é inclusivo
- 2- Ao realizar pesquisa colocando ao menos **o dia seguinte** à data de **checkout** da reserva pesquisada, ele **não retorna** a mesma com sucesso, o que demonstra que este parâmetro é o problema
- 3- O parâmetro **firstname** funciona adequadamente contanto que as observações acima sejam seguidas

Evidência:

Em virtude do parâmetro **checkin** não ser inclusivo, não há nenhum retorno para a pesquisa, pois não há nenhum outro booking que tenha um **checkin** ao menos um dia posterior ao parametrizado, assim como que contemple também os outros dois parâmetros envolvidos.

```
1 expectation failed.
JSON path size() doesn't match.
Expected: a value greater than <0>
Actual: 0
```

Cenário relacionado: C0003.

Severidade: MÉDIA

Prioridade: 2

@ResetLevel2 @TestesAPI @AcceptanceTests

B0004 – Status Code e retorno incorretos ao realizar pesquisa com filtro inválido

Método GET está retornando Status Code <200> para pesquisa inválida e retornando a lista com todos os bookings cadastrados

Passos para reproduzir:

- 1 – Preencher qualquer parâmetro inválido (não presente na schema de um objeto booking) e designar qualquer valor para ele
- 2- Realizar GET Request para "https://treinamento-api.herokuapp.com/**booking**"
- 3 – Solicitação é executada, retornando Status Code <200> e lista contendo todos os objetos "**bookings : id**"

Resultado esperado:

A pesquisa deve informar Status Code <500> e não retornar a lista contendo todos os bookings cadastrados

Observações:

- 1- Como parâmetro, qualquer valor que não faça parte da schema de um booking deve causar o problema
- 2- Outro Status Code que poderia ser utilizado, poderia ser, segundo o *Internet Engineering Task Force (IETF)*, o <400>:

400 Bad Request

The 400 (Bad Request) status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

Considerando que a solicitação contém um erro de sintaxe, malicioso ou não, do usuário

Evidência:

Validando Status Code <500> na pesquisa com parâmetro inválido

```
1 expectation failed.
Expected status code <500> but was <200>.
```

Cenário relacionado: C0004.

Severidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @AcceptanceTests @SecurityTests

B0005 – Status Code incorreto ao tentar alterar booking existente

Método PUT está retornando Status Code <403> para alteração de Booking existente quando não há autenticação do usuário

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json*
- 2- Preencher header com *Accept = application/json*
- 3- Preencher *body* com payload válido e completo
- 3- Realizar PUT Request para "https://treinamento
api.herokuapp.com/**booking/:id_existente**"

- 4- Solicitação é executada, retornando Status Code <403> e a alteração do booking não é realizada

Resultado esperado:

A pesquisa deve informar Status Code <401> e não realizar a atualização do booking

Observações:

- 1- Considerando que no momento da tentativa de alteração, o usuário não está autenticado, o Status Code <401> estaria em maior conformidade com a norma, de acordo com o *Internet Engineering Task Force (IETF)*:

401 Unauthorized

The 401 (Unauthorized) status code indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. The server generating a 401 response MUST send a WWW-Authenticate header field (Section 4.1) containing at least one challenge applicable to the target resource.

Evidência:

Validando Status Code <401> na alteração não-autenticada de um booking

```
1 expectation failed.  
Expected status code <401> but was <403>.
```

Cenário relacionado: C0005.

Severidade: BAIXA

Prioridade: 3

@ResetLevel2 @TestesAPI @AcceptanceTests

B0006 - Status Code incorreto ao tentar alteração em booking inexistente

Método PUT está retornando Status Code <405> para alteração de Booking inexistente

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json*
- 2- Preencher header com *Accept = application/json*
- 3- Preencher *body* com payload válido e completo
- 4- Preencher *Cookie* com *token válido de autenticação*
- 5- Realizar PUT Request para "https://treinamento api.herokuapp.com/**booking/id_existente**"
- 6- Solicitação é executada, retornando Status Code <405> e não realizando nenhuma alteração

Resultado esperado:

A tentativa de alteração deve retornar Status Code <404>

Observações:

- 1- O Status Code <404> poderia adequar melhor essa situação pois o recurso requisitado não existe, de acordo com o *Internet Engineering Task Force (IETF)*:

404 Not Found

The 404 (Not Found) status code indicates that the origin server did not find a current representation for the target resource or is not willing to disclose that one exists. A 404 status code does not indicate whether this lack of representation is temporary or permanent;

Evidência:

Validando Status Code <404> na alteração de um Booking não existente

```
1 expectation failed.
Expected status code <404> but was <405>.
```

Cenário relacionado: C0006.

Severidade: BAIXA

Prioridade: 3

@ResetLevel2 @TestesAPI @SmokeTests @SecurityTests

B0007 - Alteração de booking mal sucedida utilizando Basic Auth

Método PUT está retornando Status Code <403> e não efetivando a alteração de booking cadastrado

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json, Accept = application/json*
- 2- Preencher header com
Authorisation = Basic YWRtaW46cGFzc3dvcmQxMjM=
- 3- Preencher body com payload válido e completo
- 4- Realizar PUT Request para
"https://treinamento api.herokuapp.com/booking/:id_existente"
- 5- Solicitação é executada, retornando Status Code <403> e não efetuando a alteração do booking especificado

Resultado esperado:

A solicitação deve retornar Status Code <200> e realizar a alteração do booking especificado.

Observações:

- 1- Na documentação, a seção “**Booking - UpdateBooking**” está incompleta/incorreta para “JSON example usage”, tendo um header *Cookie* e não um *Authorisation*
- 2- Os exemplos para XML e URLEncoded contém a informação para *Authorization*
- 3- Na documentação, na tabela **Header**, A coluna **Description** para a linha **Authorisation** informa o valor “YWRtaW46cGFzc3dvcmQxMjM=”, o último caractere, “=”, não faz parte do valor correto
- 4- O parâmetro **Authorisation** está diferente entre a documentação e a aplicação, sendo nesta última **Authorization** a redação correta.
- 5- Realizando as correções e utilizando o header:
Authorization : Basic YWRtaW46cGFzc3dvcmQxMjM= , a validação funciona corretamente, como visto em teste adicional intitulado “*Alterar uma reserva usando o Basic auth corrigido*”

Evidência:

Validando Status Code <200> para criação de booking sem autenticação

```
1 expectation failed.
Expected status code <200> but was <403>.
```

Cenário relacionado: C0007.

Severidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @SmokeTests @SecurityTests

B0008 – Criação de booking bem sucedida para usuário não-autenticado

Método POST está retornando Status Code <200> e realização a criação de Booking para usuário não autenticado

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json*
- 2- Preencher header com *Accept = application/json*

- 3- Preencher *body* com payload válido e completo
- 4- Realizar POST Request para "https://treinamento.api.herokuapp.com/**booking**"
- 5- Solicitação é executada, retornando Status Code <200> e realizando criação de novo booking

Resultado esperado:

A solicitação deve retornar Status Code <401> e não realizar criação de novo booking

Observações:

- 1- Usuários não autenticados não devem criar bookings pois estes não ficaram atrelados a uma conta específica
- 2- O Status Code <401> deve ser utilizada para manter conformidade com o *Internet Engineering Task Force (IETF)*:

401 Unauthorized

The 401 (Unauthorized) status code indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. The server generating a 401 response MUST send a WWW-Authenticate header field (Section 4.1) containing at least one challenge applicable to the target resource.

Evidência:

Validando Status Code <401> para criação de booking sem autenticação

```
1 expectation failed.  
Expected status code <401> but was <200>.
```

Cenário relacionado: C0008.

Severidade: ALTA

Prioridade: 1

@ResetLevel2 @TestesAPI @AcceptanceTests @ContractTests @SecurityTests

B0009 - Criação de reserva bem sucedida com um parâmetro adicional

Método POST está retornando Status Code <200> e efetivando a criação de novo booking

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json*
- 2- Preencher header *Accept = application/json*

- 3- Preencher header *Cookie* com token válido de autenticação
- 4- Preencher *body* com payload completo com um parâmetro adicional ("language" : "portuguese")
- 5- Realizar POST Request para "https://treinamento api.herokuapp.com/**booking**"
- 6- Solicitação é executada, retornando Status Code <200> e efetuando a criação do novo booking

Resultado esperado:

A solicitação deve retornar Status Code <400> e não realizar a criação do novo booking

Observações:

- 1- O novo booking criado contém apenas os parâmetros válidos para a schema, ignorando todos os adicionais
- 2- Considerando que a sintaxe do objeto está correta, somente a quantidade de informação que não, o Status Code <422> pode possivelmente ser utilizado

Evidência:

Validando Status Code <400> para criação de booking com parâmetro adicional

```
1 expectation failed.  
Expected status code <400> but was <200>.
```

Cenário relacionado: C0009.

Severidade: BAIXA

Prioridade: 2

@ResetLevel2 @TestesAPI @SmokeTests @SecurityTests

B0010 - Deleção de booking efetuada com sucesso, porém com status code incorreto

Método DELETE está retornando Status Code <201> e efetivando a deleção booking cadastrado

Passos para reproduzir:

- 1- Preencher header com *Content-Type* = *application/json*
- 2- Preencher header *Cookie* com token válido de autenticação
- 3- Realizar DELETE Request para "https://treinamento api.herokuapp.com/**booking**"
- 4- Solicitação é executada, retornando Status Code <201> e efetuando a criação do novo booking

Resultado esperado:

A solicitação deve retornar Status Code <204> e realizar a deleção do booking especificado

Observações:

- 1- O Status Code atual <201>, reflete a criação de um novo recurso na aplicação, o que é o contrário do realizado aqui
- 2- O Status Code <204> também reflete a validação bem sucedida de uma ação e informa que não é retornado, como neste evento de deleção

Evidência:

Validando Status Code <204> para deleção de booking cadastrado

```
1 expectation failed.  
Expected status code <204> but was <201>.
```

Cenário relacionado: C0010.

Severidade: BAIXA

Prioridade: 3

@ResetLevel2 @TestesAPI @AcceptanceTests

B0011 - Deleção de booking efetuada com sucesso, porém com status code incorreto

Método DELETE está retornando Status Code <201> e efetivando a deleção booking cadastrado

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json*
- 2- Preencher header *Cookie* com token válido de autenticação
- 3- Realizar DELETE Request para
"https://treinamento api.herokuapp.com/**booking/-1**"
- 4- Solicitação é executada, retornando Status Code <405>

Resultado esperado:

A solicitação deve retornar Status Code <404> e informa ausência do booking requerido

Observações:

- 1- O Status Code atual <405>, enquanto válido, uma vez que nenhum método pode ser requerido em um booking inválido, pode não ser o mais específico nesta situação
- 2- O Status Code <404> informa que o pedido, enquanto válido, não pode ser executado pois a página não existe no sistema.

Evidência:

Validando Status Code <404> para deleção de booking cadastrado

```
1 expectation failed.
Expected status code <404> but was <405>.
```

Cenário relacionado: C0011.

Severidade: BAIXA

Prioridade: 3

@ResetLevel2 @TestesAPI @AcceptanceTests @SmokeTests @SecurityTests

B0012 – Criação de bookings consecutivos bem sucedida sem limitações

Método POST está retornando Status Code <200> e não limitando o usuário independente do número de POST Requests enviados

Passos para reproduzir:

- 1- Preencher header com *Content-Type = application/json*
- 2- Preencher header com *Accept = application/json*
- 3- Preencher header Cookie com informação válida
- 6- Preencher *body* com payload válido e completo
- 7- Realizar POST Request para "https://treinamento
api.herokuapp.com/**booking**"
- 8- Solicitação é executada, retornando Status Code <200> e realizando criação de novos bookings indefinidamente

Resultado esperado:

A solicitação deve retornar Status Code <429> após a criação de 2 bookings em um curto espaço de tempo

Observações:

- 1- Para fins didáticos e de facilidade no uso da API, entendo que está limitação não está presente, mas quando questionado em monitoria, o Max mencionou **2** como o valor para este teste, então realizei a validação de forma semelhante

2- O Status Code <429> parece ser o que condiz melhor com a norma neste caso, sendo o Internet Engineering Task Force (IETF):

429 Too Many Requests

The 429 status code indicates that the user has sent too many requests in a given amount of time ("rate limiting").

The response representations SHOULD include details explaining the condition, and MAY include a Retry-After header indicating how long to wait before making a new request.

Evidência:

Validando Status Code <429> após a criação de três bookings consecutivos

```
1 expectation failed.  
Expected status code <429> but was <200>.
```

Cenário relacionado: C0012.

Severidade: ALTA

Prioridade: 1

Fim do documento BDD

Considerações adicionais

- a. A capacidade de todas as pessoas acessando o recurso poderem visualizar tanto a lista de bookings quanto os bookings específicos parece ser uma falha grave de segurança, mas para fins de ensino, é necessário manter o fácil acesso. Faz sentido quando estivermos autenticados, considerando que estamos todos usando o mesmo cadastro, mas caso tivéssemos nossas próprias contas, o path **/booking/** deveria retornar somente a coleção de todos os nossos bookings pessoais, e no caso de não estarmos autenticados de nenhuma forma, retornar talvez um Status Code <401> ou mesmo um <200> e informando a necessidade de autenticação
- b. Eu imagino que uma boa quantidade das falhas nos meus testes que se referem aos Status Codes não são de fato “erros”, mas como não foi dado um número específico de problemas, talvez eu tenha ido além da conta.
- c. Eu incluí apenas os testes com falha para ficar mais resumido, se é que posso me atrever a atribuir esse adjetivo ao documento, considerando o

tamanho dele, mas tenho ciência que os cenários sem problemas também devem ser incluídos na documentação.

- d. No desafio, há duas seções para PUT e nenhuma para PATCH, apesar de estar incluído na documentação, portanto não fiz uso do recurso, apesar que teria sido útil em alguns cenários

Referências

INTERNET ENGINEERING TASK FORCE (IETF). **Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content.** Disponível em: <https://www.rfc-editor.org/rfc/rfc7231>. Acesso em: 24 out. 2021.

INTERNET ENGINEERING TASK FORCE (IETF). **Additional HTTP Status Codes.** Disponível em: <https://datatracker.ietf.org/doc/html/rfc6585>. Acesso em: 26 out. 2021.