

Desafio 03: Suíte de Testes

Objetivo

Este conjunto de testes busca avaliar o acesso, Schemas de retornos, inclusão, exclusão, pesquisa e atualização de Reservas em uma API Rest.

Estratégia

Este será um teste funcional guiado de API Rest. Serão executados testes automatizados no acesso, nas estruturas de retorno e nos fluxos de inclusão, exclusão, pesquisa e atualização de Reservas, utilizando cenários BDD (Gherkin). Os testes serão realizados em ambiente de homologação, utilizando uma máquina desktop com SO Linux Ubuntu 20.04.3 LTS, navegador Firefox 92.0, IntelliJ IDEA 2021.2.3 (Community Edition) e Allure 0.8.3.0.

Tag's

<api> <https://treinamento-api.herokuapp.com>

Testes

Suíte Health Check

Validar **sucesso** ao acessar *homepage* da <api>.

Suíte Schema

Validar **sucesso** do retorno de *Schema* da lista de Reservas;

Validar **sucesso** do retorno de *Schema* de uma Reserva específica.

Suíte Acceptance

Validar **sucesso** ao excluir Reserva;

Validar **sucesso** ao listar Id's de Reservas;

Validar **sucesso** ao listar Id's de Reservas com filtro *firstname*;

Validar **sucesso** ao listar Id's de Reservas com filtro *lastname*;

Validar **sucesso** ao listar Id's de Reservas com filtro *checkin*;

Validar **sucesso** ao listar Id's de Reservas com filtro *checkout*;

Validar **sucesso** ao listar Id's de Reservas com filtro *checkout, checkout*;

Validar **sucesso** ao listar Id's de Reservas com filtro *name, checkin, checkout*;

Validar **sucesso** ao criar uma Reserva;

Validar **sucesso** ao alterar uma Reserva usando *token*;

Validar **sucesso** ao alterar uma Reserva usando *Basic Auth*;

Suíte End to End

Validar **falha** ao excluir uma Reserva que não existe;
Validar **falha** ao excluir uma Reserva sem autorização;
Validar **sucesso** ao retornar *Status Code* 500 ao listar Id's com filtro mal formatado;
Validar **sucesso** ao retornar *Status Code* 500 quando o *Payload* for inválido;
Validar **sucesso** ao criar mais de uma Reserva em sequência;
Validar **sucesso** ao criar uma Reserva enviando *Payload* com mais parâmetros;
Validar **sucesso** ao retornar *Status Code* 418 quando o *Header Accept* for inválido;
Validar **falha** ao tentar alterar uma reserva quando o *token* não for enviado;
Validar **falha** ao tentar alterar uma reserva quando o *token* enviado for inválido;
Validar **falha** ao tentar alterar uma reserva que não existe.

Repositório de código

https://github.com/gustavoullmann/RestAssuredApiTesting/tree/Desafio_API

Cenários

Suíte Health Check

C0001 - [STATUS] (/GET) - Validar **sucesso** ao acessar *homepage* do <site>
Dado que sou um usuário externo da <api>;
Quando faço uma requisição get para a "URL" <api>/ping;
Então recebo um retorno com *Status Code* 201.
Criticidade: ALTO;
Prioridade: 1.

Suíte Schema

C0002 - [RESERVAS] (/GET) - Validar **sucesso** do retorno de *Schema* da lista de Reservas
Dado o Schema padrão de retorno da lista de Id's em formato JSON;
E a lista Id's Reservas retornada pela API;
Quando comparo os Schemas das duas listas;
Então os Schemas são equivalentes.
Criticidade: ALTO;
Prioridade: 2.

C0003 - [RESERVAS] (/GET) - Validar **sucesso** do retorno de *Schema* de uma Reserva específica

Dado o Schema padrão de retorno de uma Reserva em formato JSON;

E uma Reserva retornada pela API;

Quando comparo os Schemas das duas Reservas;

Então os Schemas são equivalentes.

Criticidade: ALTO;

Prioridade: 2.

Suíte Acceptance

C0004 - [RESERVAS] (/DELETE) - Validar **sucesso** ao excluir Reserva

Dado que sou um usuário autenticado da <api>;

E informo o Id de uma reserva válida;

Quando faço a requisição de exclusão;

Então a Reserva é deletada da lista de Id's.

Criticidade: MÉDIA;

Prioridade: 1.

Observações: o retorno de confirmação da exclusão é feito com um Status Code 201 "Created". Seria melhor usar um Status Code 200 "OK", indicando que a requisição foi bem sucedida.

C0005 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas;

Então é retornada uma lista com todos os Id's cadastrados.

Criticidade: MÉDIA;

Prioridade: 1.

C0006 - [RESERVAS] (/GET) - Validar **sucesso** ao listar uma reserva específica

Dado que sou um usuário externo da <api>;

Quando faço a requisição de uma Reserva específica usando um Id válido;

Então é retornada a Reserva solicitada.

Criticidade: ALTA;

Prioridade: 2.

C0007 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas com filtro *firstname*

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando um nome como filtro;

Então é retornada uma lista com todos os Id's cadastrados que possuem o nome.

Criticidade: MÉDIA;

Prioridade: 1.

C0008 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas com filtro *lastname*

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando um sobrenome como filtro;

Então é retornada uma lista com todos os Id's cadastrados que possuem o sobrenome.

Criticidade: MÉDIA;

Prioridade: 1.

C0009 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas com filtro *checkin*

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando a data de checkin como filtro;

Então é retornada uma lista com todos os Id's cadastrados que possuem data igual ou posterior a data informada.

Criticidade: MÉDIA;

Prioridade: 1.

C0010 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas com filtro *checkout*

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando a data de checkout como filtro;

Então é retornada uma lista com todos os Id's cadastrados que possuem data igual ou posterior a data informada.

Criticidade: MÉDIA;

Prioridade: 1.

Observações: a data deveria ser igual ou *anterior* a informada.

C0011 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas com filtro *checkout, checkout*

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando as datas de checkin e checkout como filtro;

Então é retornada uma lista com todos os Id's cadastrados que possuem datas iguais ou posteriores às datas informadas.

Criticidade: MÉDIA;

Prioridade: 1.

Observações: no caso do checkout a data deveria ser igual ou *anterior* a informada.

C0012 - [RESERVAS] (/GET) - Validar **sucesso** ao listar Id's de Reservas com filtro *name, checkin, checkout*

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando nome e as datas de checkin e checkout como filtro;

Então é retornada uma lista com todos os Id's cadastrados que possuem nome e datas iguais ou posteriores às datas informadas.

Criticidade: MÉDIA;

Prioridade: 1.

Observações: no caso do checkout a data deveria ser igual ou *anterior* a informada.

C0013 - [RESERVAS] (/POST) - Validar **sucesso** ao criar uma nova Reserva

Dado que sou um usuário externo da <api>;

E envio todas as informações obrigatórias em um *Payload* válido;

Quando faço a requisição de criar uma Reserva;

Então a Reserva é criada com sucesso.

Criticidade: ALTO;

Prioridade: 1.

Observações: apenas usuários **autenticados** devem conseguir criar Reservas.

C0014 - [RESERVAS] (/PUT) - Validar **sucesso** ao alterar uma Reserva usando *token*

Dado que sou um usuário autenticado através de *token*;

E envio novas informações em um *Payload* válido;

E informo o Id de uma Reserva válida;

Quando faço a requisição para atualizar a Reserva;

Então a Reserva é atualizada.

Criticidade: ALTO;

Prioridade: 2.

C0015 - [RESERVAS] (/PUT) - Validar **sucesso** ao alterar uma Reserva usando *Basic Auth*

Dado que sou um usuário autenticado através de *Basic Auth*;

E envio novas informações em um *Payload* válido;

E informo o Id de uma Reserva válida;

Quando faço a requisição para atualizar a Reserva;

Então a Reserva é atualizada.

Criticidade: ALTO;

Prioridade: 2.

Suíte End to End

C0016 - [RESERVAS] (/DELETE) - Validar **falha** ao excluir uma Reserva que não existe

Dado que sou um usuário autenticado da <api>;

E envio novas informações em um *Payload* válido;

E informo o Id de uma Reserva inexistente;

Quando faço a requisição para atualizar a Reserva;

Então é retornado o *Status Code* 405.

Criticidade: ALTO;

Prioridade: 3.

Observações: na documentação da API consta "*Authorisation*" onde deveria ser "*Authorization*". E a senha informada tem um "]" a mais no final.

C0017 - [RESERVAS] (/DELETE) - Validar **falha** ao excluir uma Reserva sem autorização

Dado que sou um usuário não autenticado da <api>;

E informo o Id de uma Reserva válida;

Quando faço a requisição para exclusão da Reserva;

Então é retornado o *Status Code* 403 e a Reserva **não é** excluída.

Criticidade: ALTO;

Prioridade: 1.

C0018 - [RESERVAS] (/GET) - Validar **sucesso** ao retornar *Status Code* 500 ao listar Id's com filtro mal formatado

Dado que sou um usuário externo da <api>;

Quando faço a requisição pela lista de Id's das Reservas usando um filtro mal formatado;

Então é retornado o *Status Code* 500 e a lista **não é** gerada.

Criticidade: MÉDIA;

Prioridade: 1.

Observações: a requisição retorna *Status Code* 200 ao invés do *Status Code* 500.

C0019 - [RESERVAS] (/POST) - Validar **sucesso** ao retornar *Status Code* 500 quando o *Payload* for inválido

Dado que sou um usuário externo da <api>;

E envio informações em um *Payload* inválido;

Quando faço a requisição de criar uma Reserva;

Então é retornado o *Status Code* 500 e a Reserva **não é** criada.

Criticidade: ALTO;

Prioridade: 1.

C0020 - [RESERVAS] (/POST) - Validar **sucesso** ao criar mais de uma Reserva em sequência

Dado que sou um usuário externo da <api>;

E crio com sucesso uma primeira Reserva

E envio todas as informações obrigatórias em um novo *Payload* válido;

Quando faço a requisição de criar uma segunda Reserva;

Então a segunda Reserva é criada com sucesso.

Criticidade: ALTO;

Prioridade: 1.

Observações: na documentação da API consta o termo “livro” ao invés de Reserva.

C0021 - [RESERVAS] (/POST) - Validar **sucesso** ao criar uma Reserva enviando *Payload* com mais parâmetros

Dado que sou um usuário externo da <api>;

E envio todas as informações obrigatórias, incluindo informações extras, em um *Payload* válido;

Quando faço a requisição de criar uma Reserva;

Então a Reserva é criada com sucesso.

Criticidade: MÉDIO;

Prioridade: 1.

C0022 - [RESERVAS] (/POST) - Validar **sucesso** ao retornar *Status Code* 418 quando o *Header Accept* for invalido

Dado que sou um usuário externo da <api>;

E envio informações em um *Payload* válido;

E informo um *Header Accept* inválido;

Quando faço a requisição de criar uma Reserva;

Então é retornado o *Status Code* 418 e a Reserva **não é** criada.

Criticidade: ALTO;

Prioridade: 1.

C0023 - [RESERVAS] (/PUT) - Validar **falha** ao tentar alterar uma reserva quando o *token* não for enviado

Dado que sou um usuário não autenticado da <api>;

E envio novas informações em um *Payload* válido;

E informo o Id de uma Reserva válida;

Quando faço a requisição para atualizar a Reserva;

Então a Reserva **não é** atualizada.

Criticidade: ALTO;

Prioridade: 2.

C0024 - [RESERVAS] (/PUT) - Validar **falha** ao tentar alterar uma reserva quando o *token* enviado for inválido

Dado que sou um usuário não autenticado da <api>;

E envio novas informações em um *Payload* válido;

E informo o Id de uma Reserva válida;

Quando faço a requisição para atualizar a Reserva;

Então a Reserva **não é** atualizada.

Criticidade: ALTO;

Prioridade: 2.

C0025 - [RESERVAS] (/PUT) - Validar **falha** ao tentar alterar uma reserva que não existe

Dado que sou um usuário autenticado da <api>;

E envio novas informações em um *Payload* válido;

E informo o Id de uma Reserva inexistente;

Quando faço a requisição para atualizar a Reserva;

Então a Reserva **não é** atualizada.

Criticidade: MÉDIO;

Prioridade: 2.

Resultados

Abaixo serão apresentadas algumas telas dos resultados obtidos para os casos de testes descritos acima:


```
[INFO]
[ERROR] Tests run: 21, Failures: 1, Errors: 0, Skipped: 0
[INFO]
[ERROR] There are test failures.

Please refer to /home/gustavo/IdeaProjects/RestAssuredApiTesting/target/surefire-reports for the individual test results.
Please refer to dump files (if any exist) [date]-jvmRun[N].dump, [date].dumpstream and [date]-jvmRun[N].dumpstream.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:01 min
[INFO] Finished at: 2021-10-27T16:30:27-03:00
[INFO] -----
Process finished with exit code 0
```

Imagem 01: resultado da execução dos cenários de testes pelo Maven.

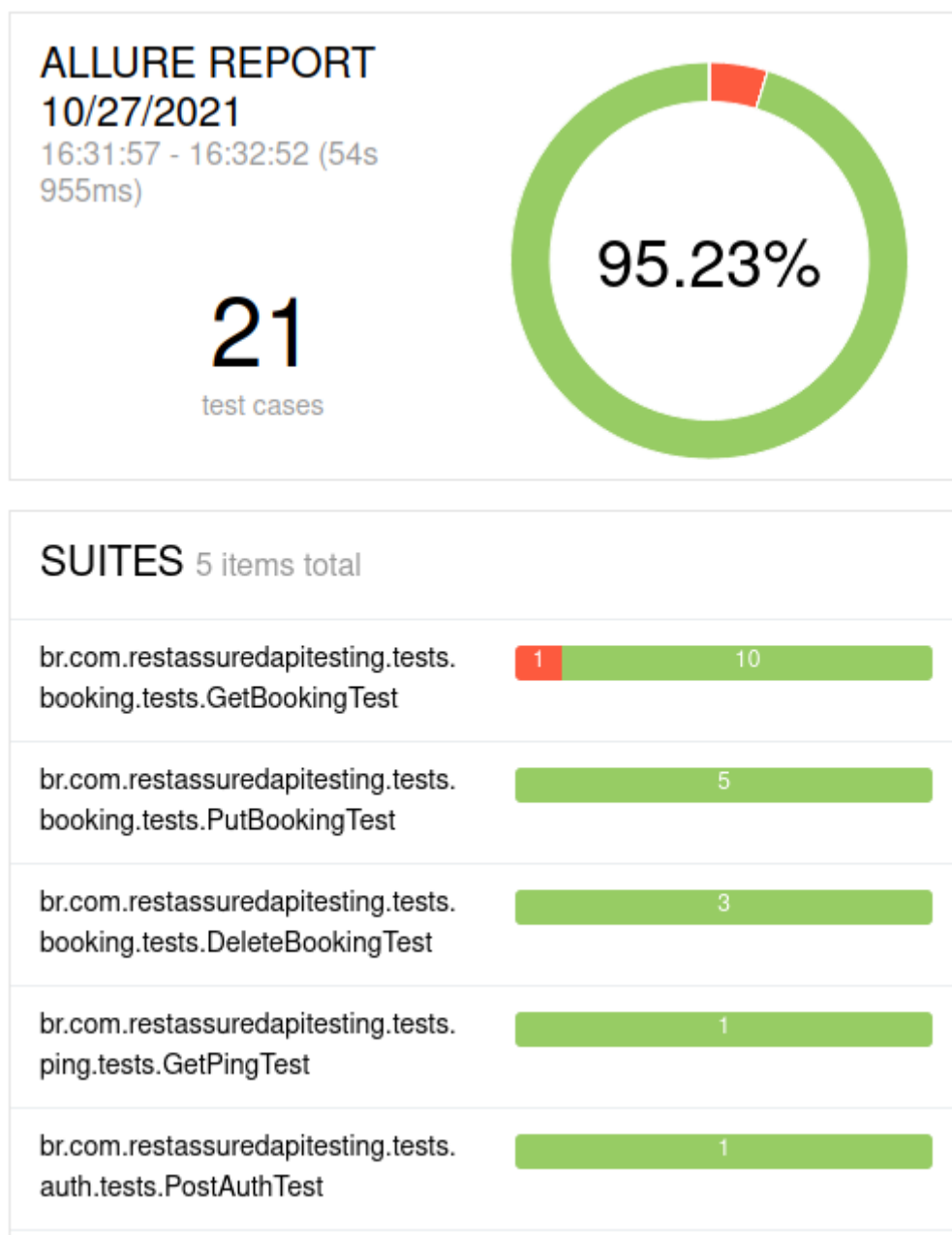


Imagem 02: resumo dos testes apresentado pelo Allure.

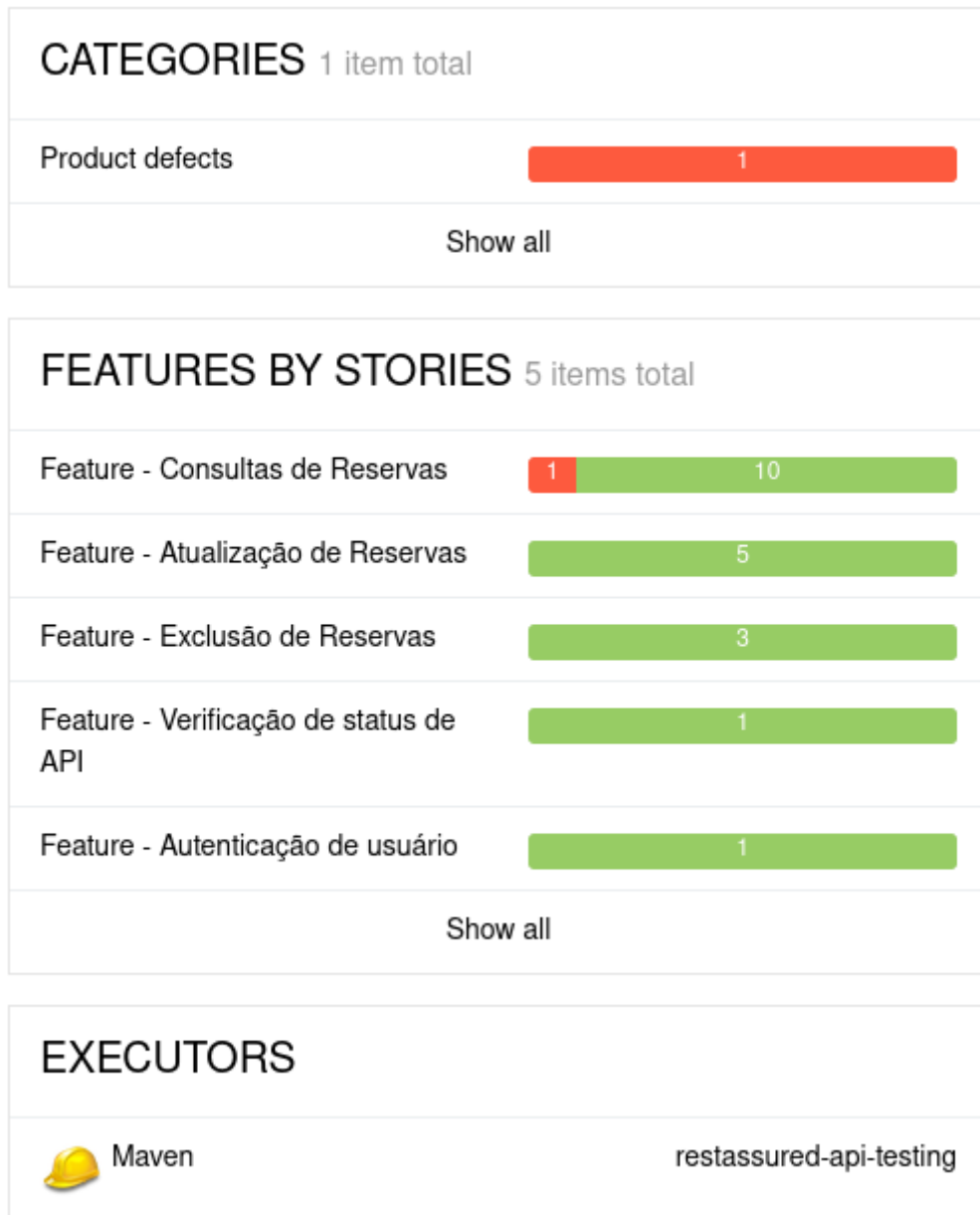


Imagem 03: resumo dos testes apresentado pelo Allure - continuação.

Failed Retornar Status Code = 500 eo executar filtro de Reservas mal formatado[Overview](#) [History](#) [Retries](#)

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

Categories: Product defects

Severity: normal

Duration: ⌚ 3s 249ms

Execution✓ **Test body**

- ✓ Retorna lista de Id's das Reservas cadastradas 1s 086ms
- Retorna uma reserva específica cadastrada usando Id 1 parameter 1s 116ms
- Retorna lista de Id's das Reservas cadastradas usando filtros 6 parameters 1s 022ms

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

Imagem 04: detalhamento do teste que falhou.