

VEM SER

Módulo 4 - Testes de API REST REST Assured (Parte 2)



DBC

Aula 5: REST Assured (Parte 2)





Aula 4 - Conteúdo Programático

4.1 Revisão

4.2 Serialização e Desserialização

4.1.1 Serialização

4.1.2 Desserialização

4.1.3 Abordagem

4.3 Classe Assertions

4.1 Revisão



4.1 Revisão: Fluxo de testes - Pipeline

1º Health Check

Garantir que o endpoint está respondendo

2º Contrato

Garantir que o endpoint não teve seus atributos alterados

3º Funcional

Garantir que o endpoint funciona ou apresenta os resultados de falhas esperados

4º Aceitação

Garantir que um conjunto de endpoints funcionam como na UI

4.1 Revisão: O que testar?

Existem vários aspectos que podem ser testados em uma API para garantir seu bom funcionamento, **por exemplo:** status, performance, tratamento de erro, segurança, etc.

Para termos **um guia**, um roadmap para identificar o que testar vamos nos utilizar da **heurística VADER**.



4.1 Revisão: O que testar?



Verbs (*Verbos*)

Authorization (*Autorização*)

Data (*Dados*)

Errors (*Erros*)

Responsiveness (*Cap. de resposta*)

4.1 Revisão: O que testar?

Verbos



Consiste em testar os métodos aptos e não aptos para o endpoint.

- GET
- POST
- PUT
- DELETE

4.1 Revisão: O que testar?

Autorização



- Validar Token (Tipo de criptografia e testes de segurança)
- Quais recursos a API Key deve acessar.
- Testes de Token, API Key ou Usuário e senha inválido ou inexistentes.
- Restrições de acesso assim que for autorizado.

4.1 Revisão: O que testar?

Dados



Observamos os dados de uma requisição ou de uma resposta de um endpoint.

- Tipagem
- Paginação
- Formato (JSON/XML)
- Schema
- Syntaxe
- Tamanho

4.1 Revisão: O que testar?

Erros



Avaliamos o status code para cada erro e suas respectivas mensagens.

- Tratamento de erros
- Detecção de erros
- Mensagens

4.1 Revisão: O que testar?

Capacidade de Resposta



- Tempo de resposta e padrão para o projeto.
- Falha rápida
- Concorrência

4.1 Revisão: O que testar?

Capacidade de Resposta



- Tempo de resposta e padrão para o projeto.
- Falha rápida
- Concorrência

4.1 Bônus: O que testar?



Quality
Tree
Software

Test Heuristics Cheat Sheet
Data Type Attacks & Web Tests

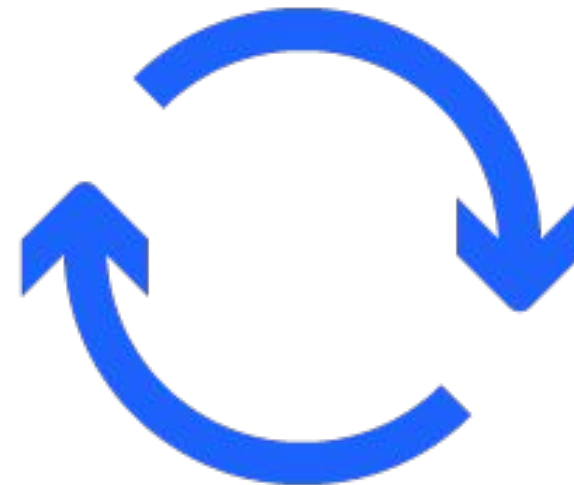
4.2 Serialização e Desserialização



4.2.1 Serialização

O que é?

No contexto do Rest Assured, serialização é o processo de **conversão de um objeto Java em um Request body (Payload)**.



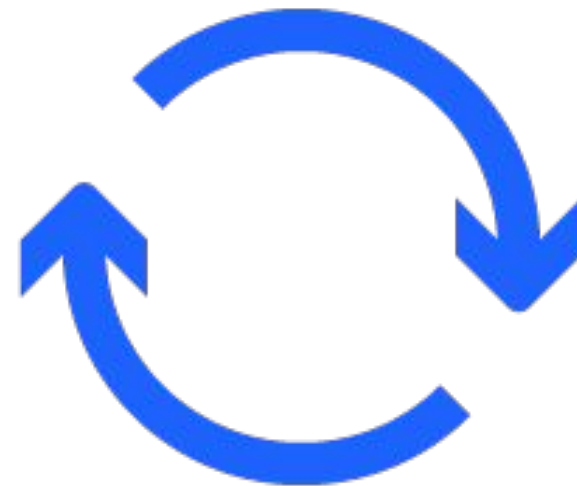
4.2.2 Desserialização

O que é?

No contexto do Rest Assured, é o processo de **conversão de um Response Body de volta em um objeto Java**.

Vantagens

- Fácil de analisar e extrair valores de resposta se eles forem agrupados como objeto Java.
- Métodos fáceis de usar podem ser criados, o que torna o código mais legível.



4.2.3 Abordagem

- O objeto Java é construído com suporte da Classe POJO
- As Classes POJO são criadas com base no Request/Response Payload.
- **gist serialization**



4.2.4 Dependências

Jackson Databind >> 2.14.2

- <https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind/2.14.2>

Gson >> 2.8.9

- <https://mvnrepository.com/artifact/com.google.code.gson/gson/2.8.9>

Classe POJO - Hands On

Desafio_POJO

- <https://gist.github.com/alysoncampos/0ff3774307b8ca15d5743efb93c7bde6>



4.2 Classe Assertions



4.2.1 Classe Assertions

No JUnit 5, Assertions é uma coleção de métodos utilitários que permitem verificar se os resultados dos testes estão corretos.

Caso a condição especificada na **asserção não seja atendida**, o teste **falhará**, lançando um **AssertionFailedError** ou uma subclasse dele.

Com eles podemos verificar se os valores são **iguais**, se são **verdadeiros ou falsos**, se estão **dentro de um determinado intervalo**, etc.

4.2 Classe Assertions



Para saber mais:

- <https://junit.org/junit5/docs/5.0.1/api/org/junit/jupiter/api/Assertions.html>
- <https://www.baeldung.com/junit-assertions>

TASK 02

Rest Assured



Referências

<https://rest-assured.io/>

<https://github.com/rest-assured/rest-assured/wiki/GettingStarted>

<https://www.baeldung.com/rest-assured-tutorial>

<https://blog.onedaytesting.com.br/testes-de-integracao-com-rest-assured/>

<https://www.toolsqa.com/rest-assured/validate-response-status-using-rest-assured/>

<https://junit.org/junit5/docs/5.0.1/api/org/junit/jupiter/api/Assertions.html>



Let's *Tech Up Together*