

PARA: Squad Nobugs

DE: Raphael Rates

DATA: 20 de Setembro de 2025

ASSUNTO: Code Review do Framework de Testes para API Restful-Booker

1. INTRODUÇÃO

O presente documento tem como objetivo formalizar a análise de código do framework de testes para a API Restful-Booker, desenvolvido por Izadora, integrante do Squad NObugs. O code review foi conduzido por Raphael Rates, também membro do mesmo squad, com foco na avaliação da qualidade do código, conformidade com boas práticas e identificação de oportunidades de melhoria.

2. CONTEXTO DO PROJETO

O framework em análise utiliza o Robot Framework em conjunto com as bibliotecas RequestsLibrary, Collections e JSONLibrary para realizar testes automatizados na API Restful-Booker. A solução implementa um fluxo completo de operações CRUD (Create, Read, Update, Delete), incluindo autenticação e validação de respostas.

3. AVALIAÇÃO GERAL

O código demonstra uma estrutura organizada e segue as convenções estabelecidas pelo Robot Framework. A autora apresentou domínio dos conceitos fundamentais de automação de testes e API testing, com implementação adequada de sessões HTTP, gestão de tokens de autenticação e sequência lógica de testes.

4. PONTOS POSITIVOS

A implementação atual apresenta vários aspectos meritórios, incluindo a organização clara da estrutura de testes, utilização apropriada de keywords com documentação adequada, correta implementação de setup e teardown, gestão eficiente de sessões HTTP e token de autenticação, além de um fluxo coeso de testes que executa operações CRUD completas. O tratamento de erros através da verificação de status codes e validação de conteúdo também é digno de nota.

5. PONTOS DE MELHORIA

Identificaram-se oportunidades de aprimoramento em algumas áreas. Recomenda-se substituir o uso de variáveis globais por variáveis de suite para melhor escopo de testes. As credenciais de autenticação encontram-se hardcoded no código, sugerindo a externalização para variáveis de ambiente ou arquivos de

configuração. A validação de valores extraídos de JSON requer ajustes para manipulação adequada de listas. A documentação das keywords pode ser ampliada para incluir informações sobre tipos de retorno. Identificou-se ainda uma issue sintática na estruturação do dictionary bookingdates que necessita correção imediata.

6. CORREÇÕES NECESSÁRIAS

Duas correções demandam atenção prioritária. A sintaxe do dictionary bookingdates apresenta erro na sua construção atual, requerendo ajuste imediato para o formato correto. A extração do token da resposta de autenticação retorna uma lista, necessitando de ajuste no código para acesso ao elemento apropriado.

7. RECOMENDAÇÕES

Recomenda-se a aplicação das correções citadas antes da implementação em ambiente produtivo. Sugere-se ainda a implementação de tratamento de exceções mais robusto, adição de mais casos de teste para cobrir cenários de erro e a consideração de padrões de page objects para melhor organização do código.

8. CONCLUSÃO

O framework desenvolvido pela Izadora apresenta qualidade elevada e está bem estruturado, necessitando apenas de ajustes pontuais para atingir excelência. O código demonstra sólido entendimento dos princípios de automação de testes e boas práticas de desenvolvimento. Com as correções recomendadas, a solução estará pronta para implantação em ambiente produtivo.

9. PRÓXIMOS PASSOS

As correções devem ser implementadas pela autora e submetidas para revalidação. Após a aplicação das mudanças, o código estará liberado para uso em produção.

Atenciosamente,

Raphael Rates

Analista de Qualidade - Squad NObugs[

Código:https://github.com/iza416/Teste_Robot1