

PeeeDrummm left a comment • edited



Code Review

- Autor: Thais Nogueira Baldino
- Revisor: Pedro Afonso de Alencar Silva
- Pull Request: [Robot API Tests #8](#)
- Branch: [robot-api-tests](#)
- Main: [Dcs-CompassUol](#)



Pontos Positivos

- Uso consistente de `Set Suite Variable` para compartilhar o token entre os testes.
- Mensagens de log bem posicionadas, que auxiliam na depuração (`Log Token gerado: ...`).
- Boa cobertura de operações CRUD (POST, GET, PUT, PATCH, DELETE).
- Uso adequado das palavras-chave `Set To Dictionary` e `Create Dictionary` .
- Boa separação de responsabilidades: os testes chamam apenas as keywords reutilizáveis.
- Testes bem nomeados, com o status esperado incluso no nome (ex: "com sucesso 200", "com sucesso 201").
- Utilização de tags por verbo HTTP facilita filtros de execução e organização dos testes.



Pontos de Melhoria

- Acessar o token como `${json[0]['token']}` pode gerar erro caso a estrutura de resposta da API seja um objeto e não uma lista. Recomenda-se utilizar `${json['token']}` para garantir compatibilidade com a estrutura retornada atualmente.
- A linha com `${EMPTY}` na criação de reservas é sobrescrita logo em seguida por `Set To Dictionary` . Pode ser eliminada sem impacto na lógica.
- Ao utilizar o token em headers (ex: com `Cookie`), é importante validar se `${auth_token}` foi definido corretamente antes da requisição. Isso evita falhas silenciosas.
- Em `test_auth.robot` , o teste depende totalmente da keyword externa. Adicionar uma validação como `Should Not Be Empty` `${auth_token}` reforça a robustez do teste.
- Após obter uma reserva com `GET- Obter reserva por ID` , incluir uma verificação como `Should Not Be Empty` `${response.json()}` garante que a resposta realmente contenha dados válidos.