

Testes Manuais - Positivos

CT001 – Listar todos os posts

Feature: Listar posts

Scenario: Valida o retorno de 100 posts

Given que a API JSONPlaceholder está disponível

When eu faço uma requisição GET para /posts

Then o status code deve ser 200

And o retorno deve conter 100 registros

And cada registro deve ter os campos "userId", "id", "title" e "body"

CT002 – Criar post

Feature: Descreve a funcionalidade testada

Scenario: Cada post deve conter os campos obrigatórios

Given que a API JSONPlaceholder está disponível

When eu faço uma requisição POST para /posts com o payload:

"""

{

 "userId": 2,

 "title": "Criado para teste",

 "body": "Este post foi criado para teste"

}

"""

Then o status code deve ser 201

And o retorno deve conter os mesmos campos enviados

And a API não salva o post permanentemente

CT003 – Buscar post por ID válido

Feature: Buscar post por ID

Scenario: Buscar post com ID 1

Given que a API JSONPlaceholder está disponível

When eu faço uma requisição GET para /posts/1

Then o status code deve ser 200

And o campo "id" deve ser 1

And os campos "userId", "title" e "body" não devem ser nulos

CT004 – Buscar usuário por ID válido

Feature: Buscar usuário por ID

Scenario: Buscar usuário com ID 10

Given que a API JSONPlaceholder está disponível

When eu faço uma requisição GET para /users/10

Then o status code deve ser 200

And o campo "id" deve ser 10

And o campo "username" deve ser "Moriah.Stanton"

And todos os campos obrigatórios do contrato JSON devem estar presentes

CT005 – Editar post existente

Feature: Editar post existente

Scenario: Editar o título do post com ID 100

Given que a API JSONPlaceholder está disponível

When eu faço uma requisição PATCH para /posts/100 com payload:

"""

{

"title": "Título editado para teste"

}

"""

Then o status code deve ser 200

And o retorno deve conter o título atualizado

And o post original não é realmente alterado na API

Testes Manuais – Negativos

CT001 – Buscar post inexistente

Feature: Testes de API para posts

Scenario: Buscar um post inexistente deve retornar 404

Given que a API de posts está disponível

When eu faço uma requisição GET para "/posts/150"

Then o código de status da resposta deve ser 404

And o corpo da resposta deve ser um objeto vazio

CT002 – Buscar campo inexistente

Feature: Testes de API para busca de campos inexistentes

Scenario: O campo "categoria" não deve existir no post de id 1

Given que a API de posts está disponível

When eu faço uma requisição GET para "/posts/1"

Then o campo "categoria" não deve estar presente na resposta

CT003 – Buscar campo que não deve existir

Feature: Testes de API para busca de campos inexistentes

Scenario: O campo "title" não deve existir no post de id 1

Given que a API de posts está disponível

When eu faço uma requisição GET para "/posts/1"

Then o campo "title" não deve estar presente na resposta

CT004 – Buscar campo incorreto

Feature: Testes de API para usuários

Scenario: Status code é 200 e e-mail está incorreto

Given que a API de usuários está disponível

When eu faço uma requisição GET para "/users/10"

Then o código de status da resposta deve ser 200

And o corpo da resposta deve ser um objeto

And o campo "id" deve ser igual a "10"

And o campo "email" não deve ser igual a "datg2702@hotmail.com"

And o campo "username" deve ser igual a "Moriah.Stanton"