

Sprint 2 — Testes Black-Box da API

Membros do grupo TP2-G7:

Tiago Figueiredo estgv17185

Diogo Ferreira pv22586

1. Teste ao Endpoint GET /apps

Descrição do Endpoint:

->Este endpoint serve apenas para os utilizadores com a role de admin listarem todas as apps so sistema, requiere autenticação com token JWT.

Cenários de testes válidos:

->Token de admin válido ,porque o token tem a role “admin” e logo deu o resultado status 200 com o body de array das aps na bd.

->Header Accept correto ,”Accept: application/json” enviado e logo deu Content-type correto.

The screenshot shows the Insomnia client interface for a GET request to `http://localhost:3000/apps`. The response status is **200 OK** with a response time of **71 ms** and a body size of **278 B**. The interface includes tabs for Params, Body, Auth, Headers, Scripts, and Docs. The 'Scripts' tab is active, showing a list of tests. The 'Tests' tab shows 3 tests, all of which passed.

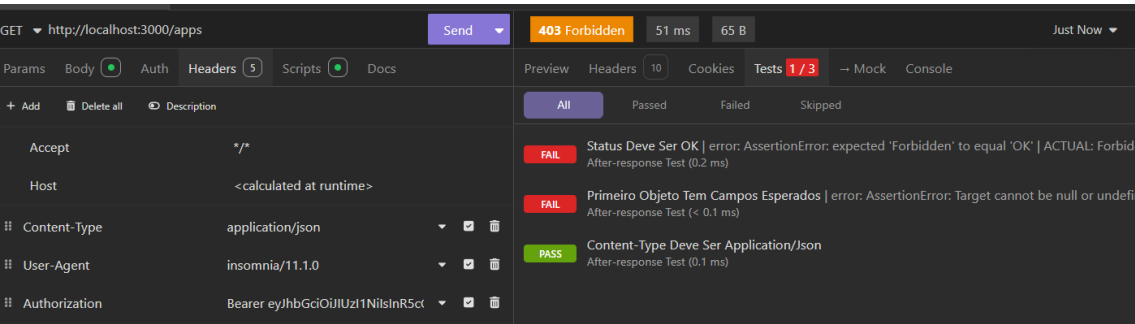
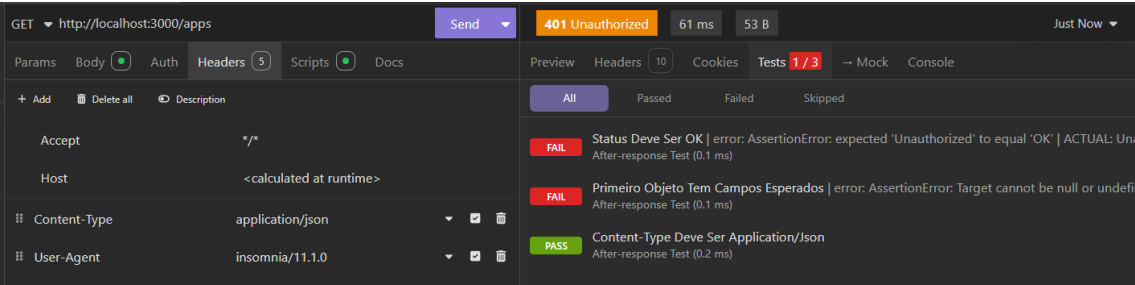
Test Name	Result	Message
Status Deve Ser OK	PASS	After-response Test (0.2 ms)
Content-Type Deve Ser Application/Json	PASS	After-response Test (0.2 ms)
Primeiro Objeto Tem Campos Esperados	PASS	After-response Test (< 0.1 ms)

```
1 - insomnia.test('Status deve ser OK', () => {
2   insomnia.expect(insomnia.response.status).toEqual("OK");
3 });
4 // Teste 2
5 insomnia.test('Content-type deve ser application/json', () => {
6   const header = insomnia.response.headers.find(
7     h => h.key.toLowerCase() === 'content-type'
8   );
9   insomnia.expect(header &&
10    header.value).to.include('application/json');
11 });
12
13 // Teste 3
14 insomnia.test('Primeiro objeto tem campos esperados', () => {
15   const data = JSON.parse(insomnia.response.body);
16   insomnia.expect(data[0]).to.have.property('id');
17   insomnia.expect(data[0]).to.have.property('nome');
18   insomnia.expect(data[0]).to.have.property('user_id');
19   insomnia.expect(data[0]).to.have.property('uuid');
20 });
```

Cenários de testes inválidos:

->Sem token ,o header authorization não é enviado ,logo dá o erro Status 401 Unauthorized.

->Token de User ,o token jwt é válido mas com a role “user”, logo resulta em status 403 forbidden



Classes de Equivalência:

Entrada	Classe Válida	Classe Inválida
Authorization	Token JWT com role: "admin"	Nenhum token / token inválido / token com role: "user"
Accept	application/json	application/xml, text/plain

Código de testes Usado:

```
insomnia.test('Status deve ser OK', () => {  
  
  insomnia.expect(insomnia.response.status).to.equal("OK");  
  
});
```

Este teste verifica se a resposta HTTP da API retorna o status "OK", representando um código HTTP 200. Se sim, confirma que o pedido foi processado com sucesso.

```
insomnia.test('Content-Type deve ser application/json', () => {  
  const header = insomnia.response.headers.find(  
    h => h.key.toLowerCase() === 'content-type'  
  );  
  insomnia.expect(header &&  
    header.value).to.include('application/json');  
});
```

Verifica se o cabeçalho Content-Type da resposta contém "application/json", indicando que o corpo da resposta está no formato JSON.

```
insomnia.test('Primeiro objeto tem campos esperados', () => {  
  const data = JSON.parse(insomnia.response.body);  
  insomnia.expect(data[0]).to.have.property('id');  
  insomnia.expect(data[0]).to.have.property('nome');  
  insomnia.expect(data[0]).to.have.property('user_id');  
  insomnia.expect(data[0]).to.have.property('uuid');  
});
```

Valida se o corpo da resposta é um array de objetos e se o primeiro objeto contém os campos esperados: id, nome, user_id e uuid.

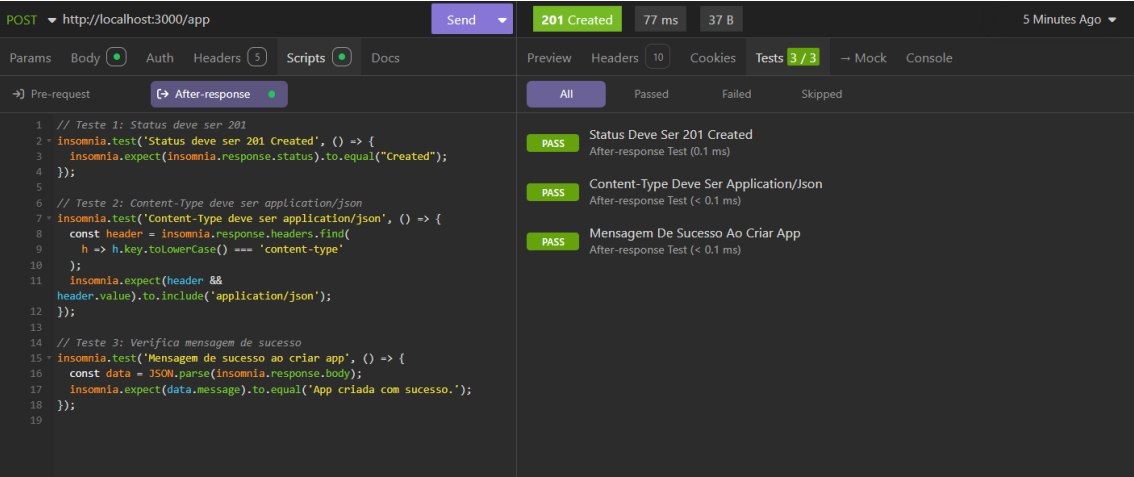
2. Teste ao Endpoint POST /app

Descrição do Endpoint:

-> Vai criar uma nova aplicação para o utilizador autenticado e recebe um JSON com o campo nome.

Cenários de testes válidos:

- >Campo nome válido , enviado “nome”.”MinhaApp”
- >Token JWT válido de admin ou user ,deixa criar a app

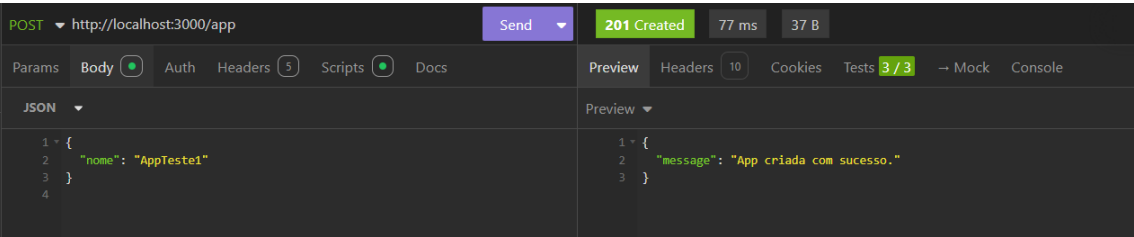


Cenários de testes inválidos:

Cenário	Descrição	Resultado
Sem Authorization	Nenhum token enviado	401 Unauthorized
Campo nome vazio	“nome”.””	400 Bad Request
Campo ausente	Body vazio{}	400 Bad Request
Content -type errado	Texto/plain	415 Unsupported Media Type

Cenários de Equivalência:

Entrada	Classe Válida	Classe Inválida
nome	Stirng não vazia	String vazia/ausente
Authorization	Jwt válido	Token inválido/ausente
Content-type	Application/json	Text/plain



Código de testes Usado:

```
insomnia.test('Status deve ser 201 Created', () => {  
  
  insomnia.expect(insomnia.response.status).to.equal("Created");  
  
});
```

Este teste valida se o status HTTP retornado é "Created", que representa o código 201, indicando que a aplicação foi criada com sucesso.

```
insomnia.test('Content-Type deve ser application/json', () => {  
  
  const header = insomnia.response.headers.find(  
  
    h => h.key.toLowerCase() === 'content-type'  
  
  );  
  
  insomnia.expect(header &&  
header.value).to.include('application/json');  
  
});
```

Verifica se a resposta HTTP possui o cabeçalho Content-Type com o valor "application/json", confirmando que o corpo da resposta está em formato JSON.

```
insomnia.test('Mensagem de sucesso ao criar app', () => {  
  
  const data = JSON.parse(insomnia.response.body);  
  
  insomnia.expect(data.message).to.equal('App criada com sucesso.');
```

});

Garante que o corpo da resposta contém uma mensagem de sucesso com o texto "App criada com sucesso.", que é o comportamento esperado da API ao criar uma nova aplicação.

3. Teste ao Endpoint PUT /password/by/:uuid

Descrição do Endpoint:

-> Este endpoint vai atualizar uma palavra passe que já existe ,pertencente a uma app de um user ,requer um token jwt , e recebe no body o novo valor da palavra passe.

Cenários de testes válidos:

->Uuid válido do próprio utilizador ,logo atualiza com sucesso.

->Palavra passe forte ,com no mínimo 8 carateres, é permitido.

Cenários de testes inválidos:

Cenário	Descrição	Resultado
UUID inexistente	Uuid não logado	404 not found
Uuid de outra pessoa	Token de user tenta acessar uuid alheio	403 forbidden
Password fraca	Ex 123	400 Bad Request
Sem token	Sem authorization	401 unauthorized

Cenários de Equivalencia:

Entrada	Classe Válida	Classe Inválida
password	String forte	String fraca
uuid	Uuid da password	Uuid inválido ou alheio
Authorization	Token valido	Ausente ou inválido

Código de testes usado:

```
insomnia.test('Status deve ser OK', () => {  
  insomnia.expect(insomnia.response.status).to.equal("OK");  
});
```

Verifica se a resposta HTTP possui status "OK", que corresponde ao código 200, indicando que a password foi atualizada com sucesso.

```
insomnia.test('Content-Type deve ser application/json', () => {  
  const header = insomnia.response.headers.find(  
    h => h.key.toLowerCase() === 'content-type'  
  );  
  insomnia.expect(header &&  
    header.value).to.include('application/json');  
});
```

Confirma que a resposta da API está no formato JSON, verificando se o cabeçalho Content-Type inclui "application/json".

```
insomnia.test('Mensagem de sucesso ao atualizar password', () => {  
  const data = JSON.parse(insomnia.response.body);  
  insomnia.expect(data.message).to.equal('Password atualizada com  
    sucesso.');
```

Verifica se o corpo da resposta JSON contém a mensagem de confirmação esperada após a atualização da password.