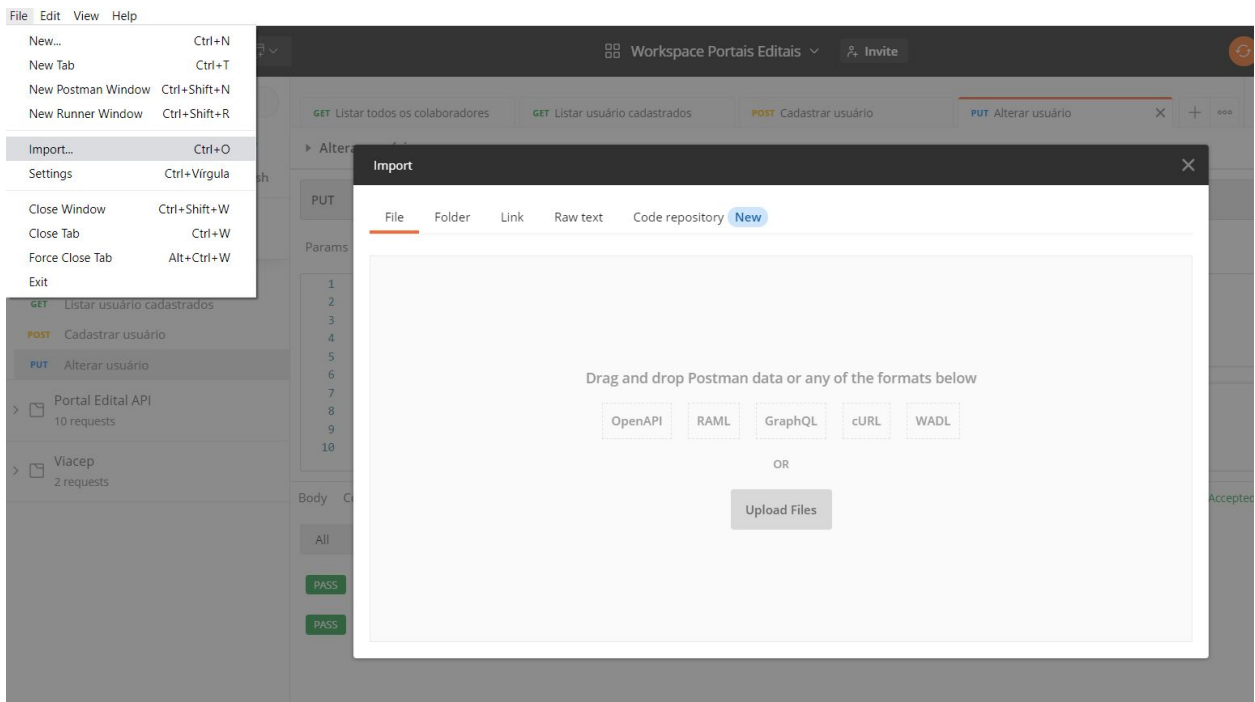


Desafio - Webservice

O exercício foi implementado utilizando o postman. Para importar, selecione no menu acima “File” > “import”, selecione o arquivo anexado “inmetrics.postman_Renato.json” para importar. A collection foi importada



Listar todos os colaboradores

Requisição GET: https://inm-test-api.herokuapp.com/empregado/list_all

Testes:

```
pm.test("Status test", function () {  
    pm.response.to.have.status(202);  
});
```

```
//verifica se os funcionários com os ids 41,42 e 43 estão listados na resposta  
pm.test("Verifica listagem funcionarios", function () {  
    pm.expect(pm.response.text()).to.include("41");  
    pm.expect(pm.response.text()).to.include("42");  
    pm.expect(pm.response.text()).to.include("43");  
});
```

The screenshot shows the Postman interface with a REST client request. The request is a GET to `https://inm-test-api.herokuapp.com/empregado/list_all`. The Tests tab is active, showing the following code:

```
1 pm.test("Status test", function () {
2   pm.response.to.have.status(202);
3 });
4
5 //verifica se os funcionários com os ids 41,42 e 43 estão listados na resposta
6 pm.test("Verifica listagem funcionarios", function () {
7   pm.expect(pm.response.text()).to.include("41");
8   pm.expect(pm.response.text()).to.include("42");
9   pm.expect(pm.response.text()).to.include("43");
10 });
```

The Test Results tab shows two failed tests:

- FAIL** Status test | AssertionError: expected response to have status code 202 but got 500
- FAIL** Body matches string | AssertionError: expected '({\"timestamp\":\"2021-03-03T21:27:35.532+0000\",\"status\":\"500\",\"error\":\"Internal Server Error\",\"message\":\"More than one record found for class: app.models.AcessoModel; nested exception is org.hibernate.HibernateException: More than one row with the given identifier was found: 30, for class: app.models.AcessoModel\",\"path\":\"/empregado/list_all\"})' to include '41'

Exibir um colaborador

Requisição GET: <https://inm-test-api.herokuapp.com/empregado/list/41>

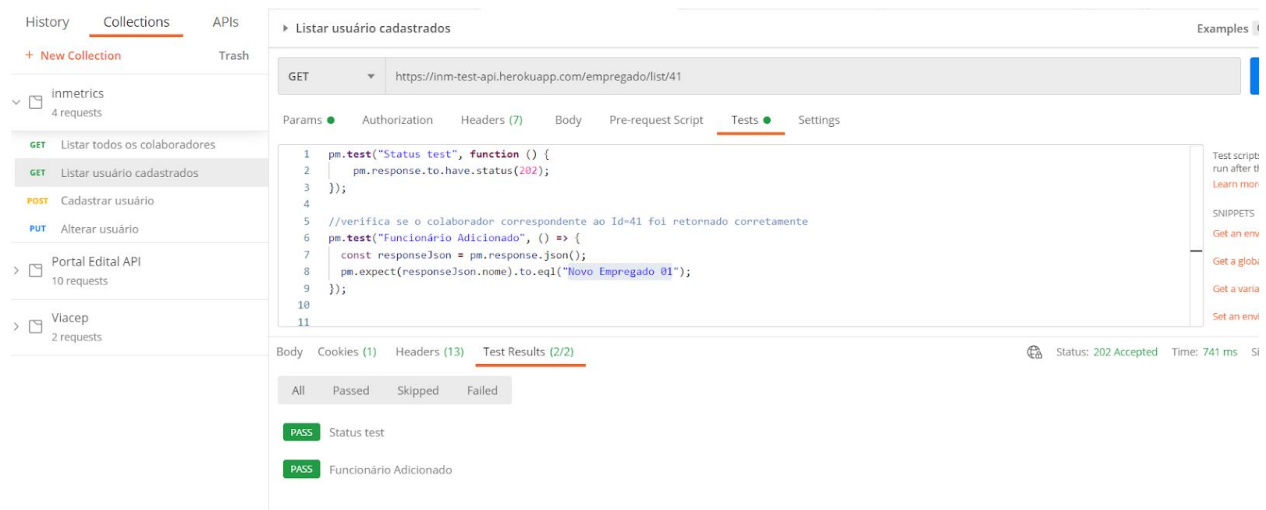
Testes:

```
pm.test("Status test", function () {
  pm.response.to.have.status(202);
});
```

```
//verifica se o colaborador correspondente ao Id=41 foi retornado corretamente
pm.test("Funcionário Adicionado", () => {
  const responseJson = pm.response.json();
  pm.expect(responseJson.nome).to.eql("Novo Empregado 01");
});
```

Resposta json:

```
{
  "empregadoId": 41,
  "nome": "Novo Empregado 01",
  "sexo": "m",
  "cpf": "388.163.430-44",
  "cargo": "Analista Editado",
  "admissao": "23/05/2021",
  "salario": "5.000,00",
  "comissao": "0,00",
  "tipoContratacao": "clt"
}
```



Cadastrar colaborador

Requisição POST: <https://inm-test-api.herokuapp.com/empregado/cadastrar>

Corpo da requisição:

```
{
  "admissao": "23/05/2021",
  "cargo": "Analista",
  "comissao": "0,00",
  "cpf": "503.873.000-07",
  "departamentoId": 1,
  "nome": "Empregado API 01",
  "salario": "4.050,00",
  "sexo": "m",
  "tipoContratacao": "clt"
}
```

Testes:

```
pm.test("Status ", function () {
  pm.response.to.have.status(202);
});

//verifica se o usuário foi cadastrado
pm.test("Funcionário Adicionado", () => {
  const responseJson = pm.response.json();
  pm.expect(responseJson.nome).to.eql("Empregado API 01");
  pm.expect(responseJson.cpf).to.eql("503.873.000-07");
});
```

Resposta json:

```
{
  "empregadoId": 54,
  "nome": "Empregado API 01",
  "sexo": "m",
  "cpf": "503.873.000-07",
```

```

    "cargo": "Analista",
    "admissao": "23/05/2021",
    "salario": "4.050,00",
    "comissao": "0,00",
    "tipoContratacao": "clt"
  }
}

```

The screenshot shows the Postman interface with a REST client request for 'Cadastrar usuário' (POST) at the URL 'https://inm-test-api.herokuapp.com/empregado/cadastrar'. The request body is a JSON object. The 'Tests' tab is active, showing a script that verifies the status and response content. The 'Test Results' section shows two passed tests: 'Status' and 'Funcionário Adicionado'.

```

1 pm.test("Status ", function () {
2   pm.response.to.have.status(202);
3 });
4
5 //verifica se o usuário foi cadastrado
6 pm.test("Funcionário Adicionado", () => {
7   const responseJson = pm.response.json();
8   pm.expect(responseJson.nome).to.eql("Empregado API 01");
9   pm.expect(responseJson.cpf).to.eql("503.873.000-07");
10 });
11
12

```

Test Results (2/2):

- PASS Status
- PASS Funcionário Adicionado

Editar colaborador

Requisição PUT: <https://inm-test-api.herokuapp.com/empregado/alterar/41>

Corpo da requisição:

```

{
  "departamentoId": 1,
  "empregadoId": 41,
  "nome": "Novo Empregado 01",
  "sexo": "m",
  "cpf": "388.163.430-44",
  "cargo": "Analista Editado",
  "admissao": "23/05/2021",
  "salario": "5.000,00",
  "comissao": "0,00",
  "tipoContratacao": "clt"
}

```

Testes:

```

pm.test("Status test", function () {
  pm.response.to.have.status(202);
});

```

```

//verifica se os parametros alterados foram editados no usuário correto
pm.test("Funcionário Alterado", () => {
  const responseJson = pm.response.json();

```

```
pm.expect(responseJson.cargo).to.eql("Analista Editado");
pm.expect(responseJson.salario).to.eql("5.000,00");
});
```

Resposta json:

```
{
  "empregadoId": 41,
  "nome": "Novo Empregado 01",
  "sexo": "m",
  "cpf": "388.163.430-44",
  "cargo": "Analista Editado",
  "admissao": "23/05/2021",
  "salario": "5.000,00",
  "comissao": "0,00",
  "tipoContratacao": "clt"
}
```

The screenshot displays the Postman interface for a PUT request titled "Alterar usuário". The URL is `https://inm-test-api.herokuapp.com/empregado/alterar/41`. The "Tests" tab is active, showing a script that performs a status test and a function test to verify the response body. The "Test Results" section at the bottom shows two tests: "Status test" and "Funcionário Alterado", both of which passed.

```
1 pm.test("Status test", function () {
2   pm.response.to.have.status(202);
3 });
4
5 //verifica se os parametros alterados foram editados no usuário correto
6 pm.test("Funcionário Alterado", () => {
7   const responseJson = pm.response.json();
8   pm.expect(responseJson.cargo).to.eql("Analista Editado");
9   pm.expect(responseJson.salario).to.eql("5.000,00");
10 });
```

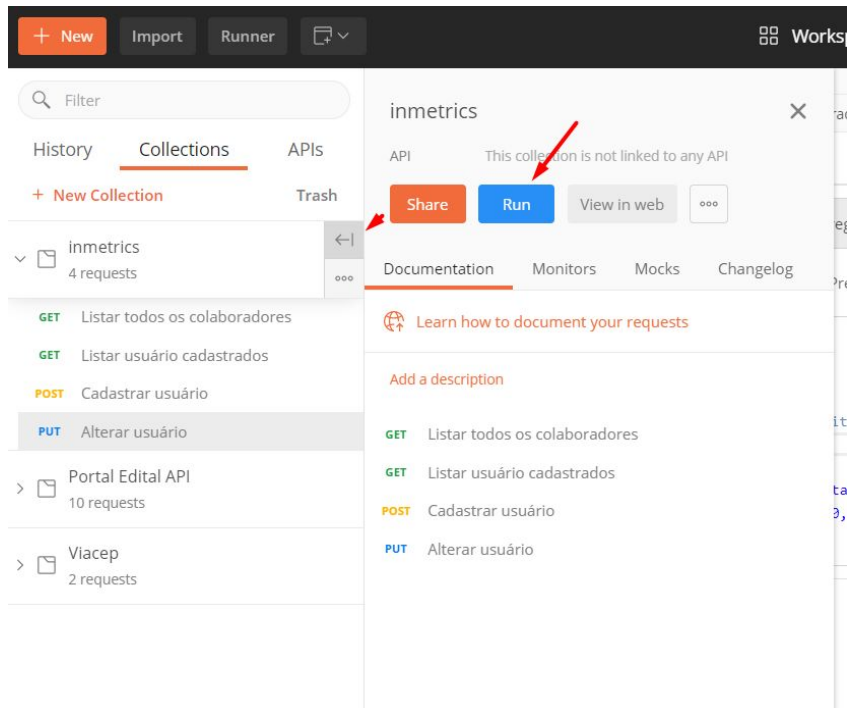
Test Results (2/2)

All	Passed	Skipped	Failed
PASS	Status test		
PASS	Funcionário Alterado		

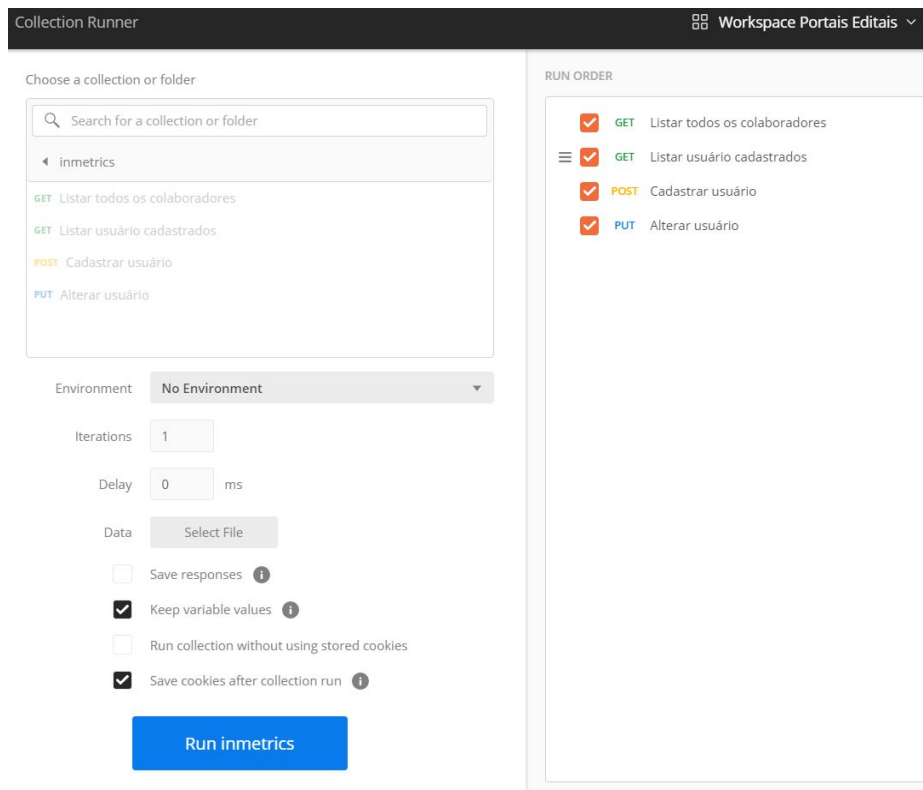
Status: 202 Accepted Time: 742 ms

Execução dos testes

Para executar o teste, selecione a collection, clique na seta menor. Um menu lateral será aberto, clique no botão run.



A aba do collection runner será aberta, clique em run inmetrics, a collection será executada.



Ao finalizar a execução dos testes, o resultado será exibido como mostrado logo abaixo, mostrando se o teste foi aprovado ou não.

Collection Runner

Run Results

Workspace Portais Editais

Run In Command Line

Docs

6 PASSED

2 FAILED

inmetrics

No Environment

just now

Run Summary

Export Results

Retry

New

Iteration 1

1

GET

Listar todos os colaboradores

https://inm-test-api.hero...

/ Listar todos os colaboradores

500 Internal Server Error

210 ms

809 B

▲

Status test

|

AssertionError: expected response to have status code 202 but got 500

Status test

|

Verifica listagem funcionarios | AssertionError: expected '{"timestamp":"2021-03-03T21:45:15.501+0000","status":500,"error":"Internal Server Error","message":"More than one row with the giv...

GET

Listar usuário cadastrados

https://inm-test-api.hero...

/ Listar usuário cadastrados

202 Accepted

190 ms

618 B

▲

Status test

Funcionário Adicionado

POST

Cadastrar usuário

https://inm-test-api.hero...

/ Cadastrar usuário

202 Accepted

200 ms

609 B

▲

Status

Funcionário Adicionado

PUT

Alterar usuário

https://inm-test-api.hero...

/ Alterar usuário

202 Accepted

204 ms

618 B

▲

Status test

Funcionário Alterado