



FGA 0238 - Testes de Software - T01 Semestre: 2024.1

Turma: T01

Nome: João Eduardo Pereira Rabelo Matrícula: 180053299

Equipe: NeverMind

Atividade 2 – Desenvolver Testes Funcionais/Sistema Caixa-Preta

1 Aplicação

Nome: MEC Energia API

Descrição: A MEC Energia API é uma aplicação desenvolvida para gerenciar e disponibilizar dados sobre consumo de energia, permitindo o monitoramento e análise dos mesmos através de endpoints de API.

Funcionalidade: Registro de Unidade Consumidora

Descrição: Esta funcionalidade permite ao usuário registrar, atualizar, listar e recuperar informações sobre unidades consumidoras associadas a universidades, com verificações de permissões de acesso baseadas no tipo de usuário e na universidade.

2 Especificação da Funcionalidade

Feita a configuração designada no REAME.txt do projeto, é feito o "docker compose up" e a aplicação já estará rodando na sua máquina.

Logo após a API estará disponível no endereço: http://localhost:8000 aonde acessaremos os endpoints do aplicação, localizados neste outro endereço: http://localhost:8000/api/.





```
Api Root

The default basic root view for DefaultRouter

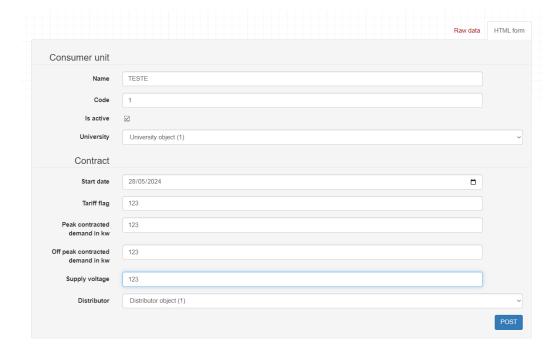
GET /api/

HITP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "universities": "http://localhost:8000/api/universities/",
    "consumer-unites": "http://localhost:8000/api/consumer-units/",
    "energy-bills": "http://localhost:8000/api/consumer-units/",
    "universities": "http://localhost:8000/api/consumer-units/",
    "university-user": "http://localhost:8000/api/consumer-units/",
    "university-user": "http://localhost:8000/api/denergy-bills/",
    "university-user": "http://localhost:8000/api/distributors/",
    "distributors": "http://localhost:8000/api/distributors/",
    "distributors": "http://localhost:8000/api/distributors/",
    "distributors": "http://localhost:8000/api/distributors/",
    "download-step-by-step-pdf": "http://localhost:8000/api/download-step-by-step-pdf/",
    "recommendation-settings": "http://localhost:8000/api/recommendation-settings/"
}
```

Neste endereço iremos acessar a funcionalidade "Consumer Units" e lá iremos fazer os testes nas seguintes funcionalidades:

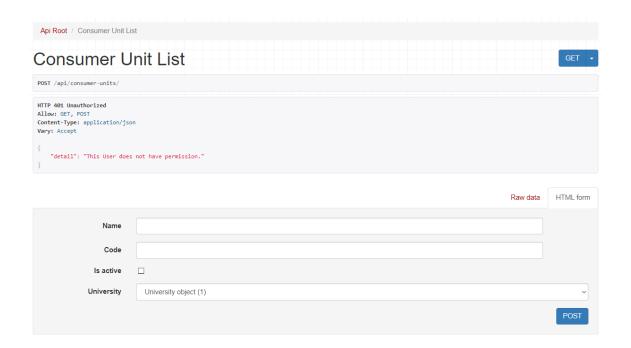
- Endpoint de criação de unidade consumidora (POST /consumer-units/)
- Endpoint de atualização de unidade consumidora (PUT /consumer-units/{id}/)



- Endpoint de listagem de unidades consumidoras (GET /consumer-units/)
- Endpoint de recuperação de unidade consumidora (GET /consumer-units/{id}/)







3 Condições de Entrada e de Saída

#	Condição de Entrada	Condição de Saída	Classe de Equivalência
1	Dados válidos de unidade consumidora.	Unidade Consumidora Criada/Atualizada.	Válida
2	Dados inválidos de unidade consumidora.	Erro de Validação.	Inválida
3	Permissões Válidas ao Usuário.	Acesso Permitido.	Válida
4	Permissões Inválidas ao Usuário.	Erro de Autorização.	Inválida
5	Filtros Válidos para Listagem.	Lista de Unidades Consumidoras.	Válida
6	Filtros Inválidos para Listagem.	Erro de Validação.	Inválida





4 Especificação dos Casos de Teste

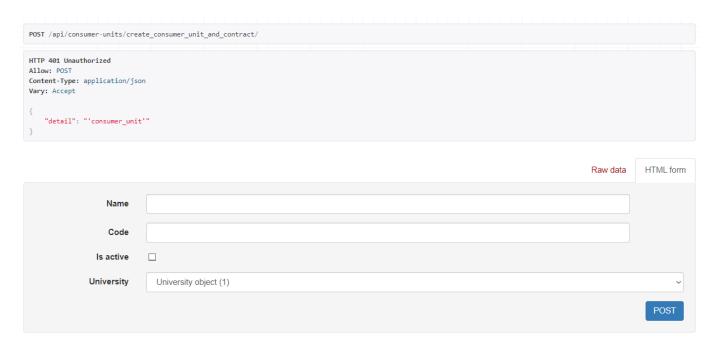
	Título do Caso de	Classes de			
#	Teste	Equivalência	Dados de Entrada	Saída Esperada	Procedimento
1	Registro de unidade consumidora com dados válidos.	1, 3	{"name": "Unit1", "university": 1}	Unidade consumidora criada com sucesso	Chamar API com dados válidos
2	Registro de unidade consumidora com dados inválidos.	2	{"name": "", "university": 1}	Erro de validação	Chamar API com nome vazio
3	Atualização de unidade consumidora com permissões válidas.	1, 3	{"name": "Unit1 Updated", "university": 1}	Unidade consumidora atualizada com sucesso	Chamar API com dados válidos e usuário com permissões
4	Atualização de unidade consumidora com permissões inválidas.	1, 4	{"name": "Unit1 Updated", "university": 1}	Erro de autorização	Chamar API com dados válidos e usuário sem permissões
5	Listagem de unidades consumidoras com filtros válidos.	3, 5	{"university_id" : 1}	Lista de unidades consumidoras	Chamar API com filtros válidos
6	Listagem de unidades consumidoras com filtros inválidos.	6	{"invalid_filter ": "invalid_value"}	Erro de validação	Chamar API com filtros inválidos

5 Execução dos Testes

• T-1 Resultado: Usuário Admin sem autorização.



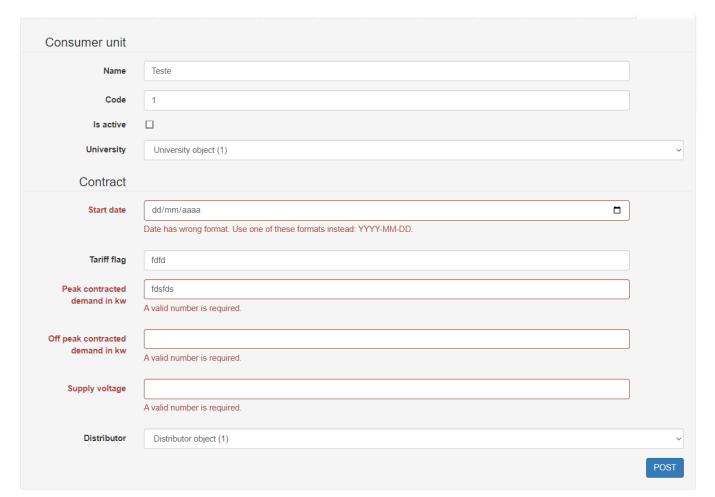




• T-2 Resultado: Retorna erro 400, e informa quais campos estão inválidos.







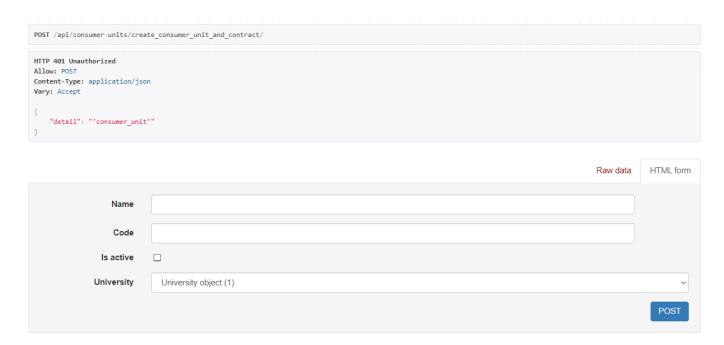
• T- 3 Resultado: Usuário Admin sem autorização.







• T- 4 Resultado: Conforme esperado, retorna que não foi autorizada a request.



T- 5 Resultado: Usuário Admin sem autorização.







T- 6 Resultado: Usuário Admin sem autorização.



6 Falhas Identificadas

Após a execução dos testes de caixa preta, na medida do que foi permitido o acesso, não foi encontrado erros ou falhas na api da mec energia.





7 Conclusão

Os testes caixa preta realizados, informaram que a ferramenta para oque foi proposto, está teoricamente funcionando de acordo como deveria, foram feitas tentivas para criação e alteração das unidades consumidoras, contudo, não foi possível dar continuidade e listar os contratos criados para as unidades devido a necessidade de permissão do usuário criado para dar continuidade.

Com base nisso, sugiro aprimorar a documentação do projeto e ajustar os casos de teste para que as ferramentas possam ser testadas de maneira interativa com permissão, tendo em vista que até no swagger, os testes destas ferramentas não apresentam o resultado esperado: http://localhost:8000/api/swagger/schema/.