**Documentação desafio TRACTIAN**

Desafio proposto: Implementar CRUD para cadastrar empresas, unidades, ativos e usuários.

Critérios:

* Cada ativo deve ter uma imagem, nome, descrição, modelo, proprietário, status e nível de integridade;
* Cada ativo faz parte de uma unidade;
* Cada unidade faz parte de uma empresa;
* Cada usuário faz parte de uma empresa;
* Existem três tipos de status: Em execução, Alerta, Parado;
* O nível de saúde precisa estar entre 0% e 100%.

Para consumir a API faz-se necessário informar no header da requisição um token de forma a assegurar o acesso à apenas quem possuir um token válido.

Nome : token

Valor : a473bf6226c4c6b6d45fc0f5926816e3

Tipo: Header

Foi implementado uma rota para cada item levantado na proposta onde as rotas possuem um prefixo (URL\_Base/api/...) e nome do item, todas as rotas utilizam os seguintes métodos [GET,POST,PUT, DELETE], para cada item quando utilizado o método GET pode ser informado no header da requisição qual parâmetro será considerado para fazer a consulta, tendo algumas regras na utilização de alguns parâmetros que será permitido a utilização de apenas um parâmetro por vez na consulta, caso seja informado mais de um que seja bloqueado pela api, será retornado a mensagem que apenas um parâmetro poderá ser utilizado para fazer o filtro ( entende-se parâmetros únicos na consulta os que forem marcados com \*, caso o campo não tenha o \* poderá ser utilizando juntamente como filtro adicional ), segue abaixo as rotas e exemplos de usa utilização:

**URL\_BASE: localhost:3000/api**

***Rota para Empresa : URL\_BASE/empresa***

GET:

Header: [cnpj \* , nome\_fantasia \*, rasao\_social \*]

POST:

body: {

"nome\_fantasia": "EMPRESA MODELO",

"rasao\_social": "EMPRESA MODELO",

"cnpj": "00.000.000/0001-00",

}

PUT:

body: {

"\_id":"(id gerado pelo mongodb) ",

"nome\_fantasia": "EMPRESA MODELO",

"rasao\_social": "EMPRESA MODELO",

"cnpj": "00.000.000/0001-00",

}

DELETE:

body: {

"\_id" : " (id gerado pelo mongodb) ",

"nome\_fantasia": "EMPRESA MODELO",

"rasao\_social": "EMPRESA MODELO",

"cnpj": "00.000.000/0001-00",

}

***Rota para Usuário: URL\_BASE/usuario***

GET:

Header : [ nome \*, login \* , empresa ]

Obs: o parâmetro empresa poderá ser passado tanto o objeto quanto apenas o seu id;

POST:

body: {

"nome": "Luiz Paulo",

"login": "lluizppaulo",

"senha": "123456",

"empresa": (Objeto da empresa / id)

}

PUT:

body: {

"\_id": " (id gerado pelo mongodb) ",

"nome": "Luiz Paulo",

"login": "lluizppaulo",

"senha": "123456",

"empresa": (Objeto da empresa / id)

}

DELETE:

body: {

"\_id": " (id gerado pelo mongodb) ",

"nome": "Luiz Paulo",

"login": "lluizppaulo",

"senha": "123456",

"empresa": (Objeto da empresa / id)

}

***Rota para Unidade: URL\_BASE/unidade***

GET:

Header : [descricao\*, endereço\*, bairro\*, ponto\_referencia\*, empresa]

Obs: o parâmetro empresa poderá ser passado tanto o objeto quanto apenas o seu id;

POST:

body: {

"descricao": "UNIDADE TESTE",

"endereco": "RUA GONÇALVES DIAS",

"bairro":"CENTRO",

"ponto\_referencia": "AO LADO DO COLEGIO DON PRADA",

"empresa": "(objeto da empresa / id )"

}

PUT:

body: {

"\_id": " (id gerado pelo mongodb) ",

"descricao": "UNIDADE TESTE",

"endereco": "RUA GONÇALVES DIAS",

"bairro":"CENTRO",

"ponto\_referencia": "AO LADO DO COLEGIO DON PRADA",

"empresa": "(objeto da empresa / id )"

}

DELETE:

body: {

"\_id": " (id gerado pelo mongodb) ",

"descricao": "UNIDADE TESTE",

"endereco": "RUA GONÇALVES DIAS",

"bairro":"CENTRO",

"ponto\_referencia": "AO LADO DO COLEGIO DON PRADA",

"empresa": "(objeto da empresa / id )"

}

***Rota para Unidade: URL\_BASE/ativo***

GET:

Header : [ nome, modelo, proprietario, status, nivel\_integridade, unidade]

Obs: o parâmetro proprietário e unidade poderá ser passado tanto o objeto quanto apenas o seu id, é importante ressaltar que todos os campos mencionados para filtro poderão ser utilizados em conjunto, ficando desta forma

POST:

body: {"imagem": "(Imagem em base64)",

"nome":"ATIVO TESTE",

"modelo":"TESTE",

"proprietario": "(objeto ou id do usuário )",

"status": "Parado",

"nivel\_integridade":100,

"unidade": "(objeto ou id da unidade)"

}

PUT:

body: {

"\_id":(id gerado pelo mongodb),

"imagem": "(Imagem em base64)",

"nome":"ATIVO TESTE",

"modelo":"TESTE",

"proprietario": "(objeto ou id do usuário )",

"status": "Parado",

"nivel\_integridade":100,

"unidade": "(objeto ou id da unidade)" ,

"alterou\_imagem": (true/false),

"link\_imagem\_anterior": (link para deletar a imagem antiga)}

DELETE:

body: {

"\_id":(id gerado pelo mongodb),

"imagem": "(Imagem em base64)",

"nome":"ATIVO TESTE",

"modelo":"TESTE",

"proprietario": "(objeto ou id do usuário )",

"status": "Parado",

"nivel\_integridade":100,

"unidade": "(objeto ou id da unidade)" ,

"alterou\_imagem": (true/false),

"link\_imagem\_anterior": (link para deletar a imagem antiga)}

}

***Rota para Histórico do Ativo : URL\_BASE/ativo/historico***

Esta rota é composta apenas pelo método GET e possui como parâmetro de consulta o campo id\_ativo, será retornado um log contendo todas as alterações já realizadas no ativo como por exemplo a troca de status mudança de nome, modelo e nível de integridade, apenas o campo de imagem não será registrado a sua alteração uma vez que ao ser alteado a imagem do ativo ela é deletada do repositório não podendo mais ser consultada.

GET:

Header: [id\_ativo]

Exemplo de retorno:

[

{

"\_id": "625ce5b60c769d1664da5f9c",

"id\_ativo": "62586b508cec50273492b74a",

"nome\_new": "ATIVO TESTE HEROKU1",

"nome\_old": "ATIVO TESTE HEROKU",

"data\_alteracao": "18/04/2022 01:14:46"

},

{

"\_id": "625ce5c00c769d1664da5f9d",

"id\_ativo": "62586b508cec50273492b74a",

"modelo\_new": "TESTE",

"modelo\_old": "TESTE 23",

"data\_alteracao": "18/04/2022 01:14:56"

},

{

"\_id": "625ce5cf0c769d1664da5f9e",

"id\_ativo": "62586b508cec50273492b74a",

"nome\_new": "ATIVO TESTE HEROKU5",

"nome\_old": "ATIVO TESTE HEROKU1",

"nivel\_integridade\_new": 90,

"nivel\_integridade\_old": 100,

"data\_alteracao": "18/04/2022 01:15:11"

}

]