Olá aqui você vai encontrar alguns exercícios simples com foco em testes funcionais de frontend e backend, e um voltado para automação de backend.

Caso possua dúvidas, não hesite em perguntar.

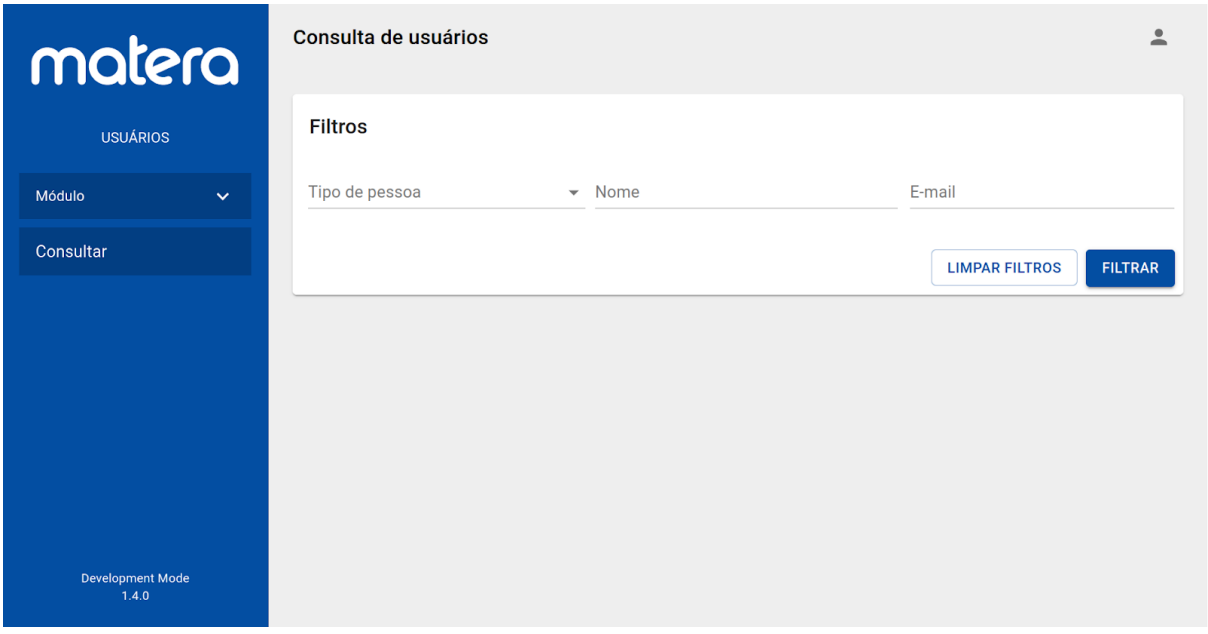
Boa sorte!

Exercício 1 - Testes funcionais de frontend

Imagine que você trabalha com um sistema básico de ERP desenvolvido para a Web. Acabou de chegar uma nova demanda para você testar na qual é a criação de uma nova tela, onde será possível consultar os usuários cadastrados neste ERP.

1. Descreva alguns casos de testes para testar apenas os campos de filtros desta tela.
2. Descreva apenas os casos principais. Pelo menos 3 caminhos felizes e 2 infelizes.
3. Os casos de testes levantados devem ser descritos no formato BDD (Behavior Driven Development).
4. Desejável que os casos sejam escritos em Inglês.

Aqui podemos observar uma imagem desta tela:

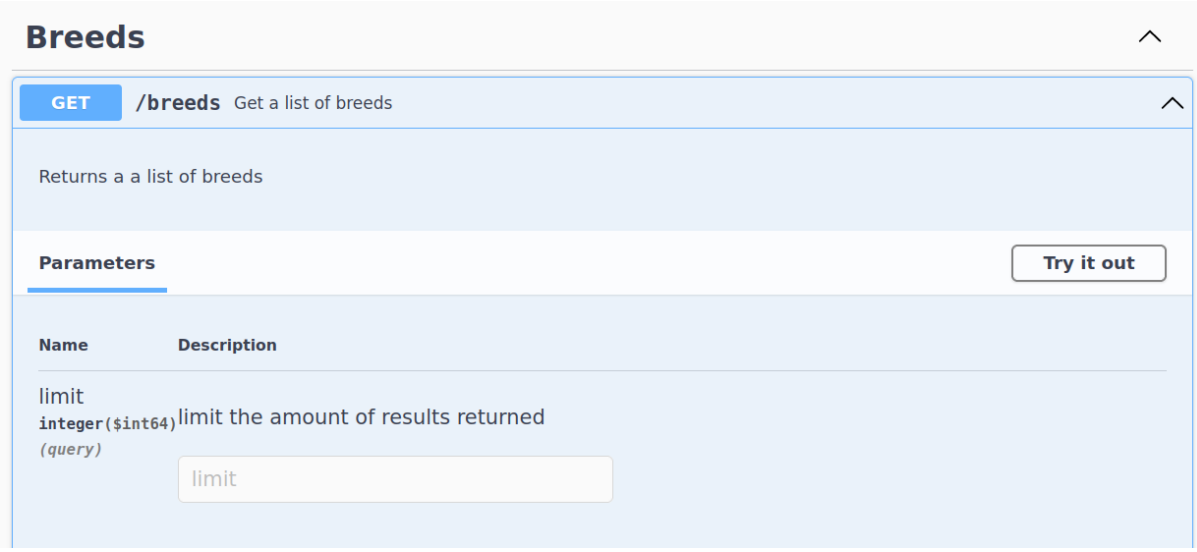


Exercício 2 - Testes funcionais de backend

Agora vamos imaginar que você trabalha em um time que provê APIs para seus clientes. Acabou de chegar uma nova demanda para você testar um novo endpoint de consulta. Aqui vamos simular este endpoint através da API pública Cat Facts API (https://catfact.ninja/), onde o endpoint a ser testado é o “breeds”. No link “https://catfact.ninja/” você poderá ver o swagger.

1. Descreva alguns casos de testes para testar o endpoint GET “/breeds”.
2. Descreva apenas os casos principais. Pelo menos 3 caminhos felizes e 2 infelizes.
3. Os casos de testes levantados devem ser descritos no formato BDD (Behavior Driven Development).
4. Desejável que os casos sejam descritos em Inglês.

Aqui podemos observar uma imagem do swagger:

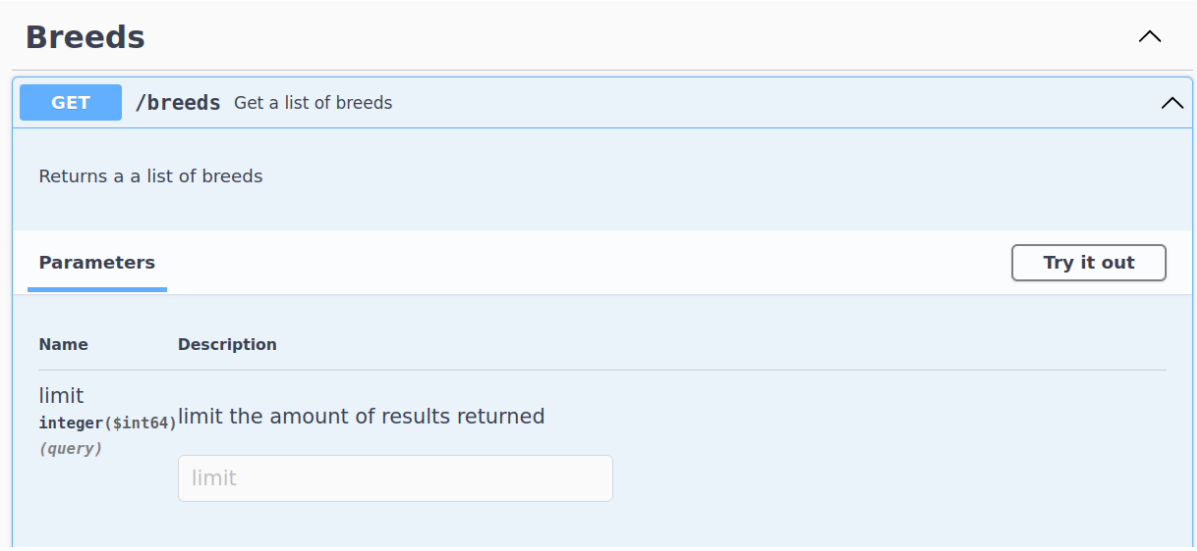


Exercício 3 - Testes automatizados

Vamos continuar no mesmo contexto do exercício anterior, onde você trabalha em um time que provê APIs para seus clientes. Na mesma demanda anterior, você vai precisar criar uma automação para o mesmo endpoint. Vamos simular este endpoint através da API pública Cat Facts API (https://catfact.ninja/), onde o endpoint a ser testado é o “breeds”. No link “https://catfact.ninja/” você poderá ver o swagger.

1. Desenvolva alguns casos de testes para testar o endpoint GET “/breeds”.
2. Desenvolva apenas os casos principais. Pelo menos 1 caminho feliz e 1 infeliz.
3. Você pode usar qualquer linguagem e framework de sua preferência.
4. Desejável utilizar o framework Cucumber para descrição dos casos.
5. Ao final, favor disponibilizar o código através do GitHub (https://github.com/).

Aqui podemos observar uma imagem do swagger:



Obrigado!