



Teste para contratação de Desenvolvedores Front-End

Parabéns por fazer parte da etapa de testes da nossa seleção, será um grande prazer analisar o seu código.

O teste basicamente tem como objetivo a avaliação de consumo da API, o design do layout não é tão importante **(mas não deixe isso limitar a sua criatividade)**. Esperamos que tenha um ótimo desempenho.

Instruções básicas

Necessário

- Usar TypeScript
- Usar ant.design <https://ant.design/>
- Boas práticas do Javascript
- Todo o código deverá ser escrito em inglês
- Colocar em algum repositório público, GitHub ou Bitbucket
- Armazenar a sessão para que tenha lógica de permanecer logado corretamente ao atualizar a página.
- Testes nos componentes serão um diferencial

Flow

- 1 - Toda requisição precisa ser feita com uma sessão, então utilize o /auth/session-request com a sua chave para obter.
- 2 - Agora com a sessão iniciada, poderá registrar um usuário nela. utilize /auth/session-register
- 3 - Após conseguir autenticar a sessão, poderá fazer busca no endpoint privado /pet/search

Prazo

Entendemos que pode ser executado somente o que contempla o teste, ou o candidato poderá turbiná-lo, e não queremos de maneira nenhuma impedir que isto ocorra. Desta forma o prazo começa a conta a partir de hoje (26/12/2019), e terminará na noite do dia 30/12/2019.

OBS

Caso consiga não utilizar o Redux será um grande diferencial, uma vez que não o utilizamos, criamos a nossa própria solução, porém, se não conseguir, não tem

problema, caso venha fazer parte da nossa equipe, teremos o prazer de passar este conhecimento.

API

Session Request

Em nossa API, todo endpoint para ser consumido, precisa enviar em seu Header a tag Authorization um Bearer token. E quem fornece este Bearer token é este primeiro endpoint básico.

Cada um do teste terá um `_API_KEY_` para conseguir consumir a API, este é o seu **505763d6-4202-4b05-9efc-93b366939bcf**, desta forma saberemos o seu comportamento em de uso da nossa API.

Endpoint

<https://test.adoptets.app/v1/auth/session-request>

Request {"system_api_key": `_API_KEY_`}

Response

Você terá como retorno um JSON, e nele terá o `access_key`, o conteúdo desta chave é o seu Bearer token, que servirá para consumir o próximo endpoint.

Session Register

Agora que tem o Bearer token em mãos, poderá registrar o usuário na sessão. Após autenticar, terá como retorno os dados deste usuário, e no final do json, terá um `access_key` novo autenticado com este usuário, este deverá ser a sua nova chave, para que possa executar o novo endpoint.

Endpoint

<https://test.adoptets.app/v1/auth/session-register>

Request

{"organization_user": {"email": "usuario-test@adoptets.com", "password": "123123"}}

Response

Você terá como retorno um JSON, e nele terá um novo `access_key`, o conteúdo desta chave é o seu Bearer token, que servirá para consumir o próximo endpoint.

Pet Search

Agora que já tem uma Bearer token autenticado, poderá fazer esta requisição, e nele obter todos os pets. Nesta etapa terá como objetivo Listar, Filtrar, Ordenar, Paginar, tudo isso neste mesmo endpoint.

OBS:

Dentro de search, terá `sex_key` o seu filtro, como exemplo. Poderá filtrar somente estes campos, o exemplo do request já está com o filtro de `sex_key` em female:

`sex_key => MALE, FEMALE`

`size_key => S, M, L, XL`

`age_key => BABY, YOUNG, ADULT, SENIOR`

Para ordenar, basta utilizar a key `sort`, que se encontra em options no json, e adicionar o campo que deseja ordenar;

exemplo para ASC de name => `"sort": ["name"]` ou `"sort": ["specie.name"]`

exemplo para DESC de name => `"sort": ["-name"]` ou `"sort": ["-specie.name"]`

Endpoint <https://test.adopets.app/v1/pet/search>

Request

```
{
  "search": {
    "sex_key": "FEMALE",
    "_fields": [
      "id",
      "uuid",
      "custom_code",
      "name",
      "specie_id",
      "breed_primary_id",
      "price",
      "created_date",
      "status_key",
      "branch_id",
      "payment_model_key",
      "sex_key",
```

```

        "size_key",
        "age_key"
    ],
    "specie": {
        "with": {
            "_fields": [
                "id",
                "name"
            ]
        }
    },
    "breed_primary": {
        "with": {
            "_fields": [
                "id",
                "name"
            ]
        }
    },
    "branch": {
        "with": {
            "uuid": "ef71cadf-fa9b-4c8b-a1a8-0e31e784c3ff",
            "_fields": [
                "id",
                "uuid"
            ]
        }
    }
},
"options": {
    "page": 1,
    "limit": 5,
    "sort": []
}
}

```

Response

Você terá como retorno um JSON, e nele terá todas as informações de paginação, e as informação de cada pet.