Documentação do Desafio Triágil

Essa documentação fornece uma visão geral do código em Python que implementa uma API para gerenciamento de equipes de Pokémon utilizando Flask.

**Funcionalidades**

Obter todas as equipes de Pokémon: A rota ./api/teams permite obter todas as equipes de Pokémon armazenadas no sistema.

Obter as equipes de um usuário específico: A rota ./api/teams/<user> permite obter as equipes de Pokémon associadas a um usuário específico.

Criar uma nova equipe de Pokémon: A rota ./api/teams permite criar uma nova equipe de Pokémon, fornecendo um nome de usuário e uma lista de nomes de Pokémon. A API valida se os Pokémon fornecidos existem na PokeAPI antes de criar a equipe.

**Requisitos**

Python 3.12

Flask

Requests

**Instalação:**

Certifique-se de ter o Python instalado. Caso contrário, você pode baixá-lo e instalá-lo a partir do site oficial. (<https://www.python.org/>)

**Instale as dependências do projeto executando o seguinte comando:**

pip install flask requests

**Utilização:**

Clone ou faça o download do repositório (<https://github.com/DavidGBraga/Desafio-Triagil-2024>) que contém o código Python.

Navegue até o diretório onde o código está localizado.

**Execute o aplicativo Flask no terminal executando o seguinte comando:**

*python pokeservices.py*

O servidor Flask será iniciado e estará acessível em <http://localhost:5000>.

**Execute o Docker no terminal executando o seguinte comando:**

*docker-compose up*

**Rotas**

Obter todas as equipes de Pokémon

**Método: GET**

Rota: */api/teams*

Descrição: Retorna todas as equipes de Pokémon armazenadas no sistema.

Obter as equipes de um usuário específico

**Método: GET**

Rota: */api/teams/<user>*

Parâmetros: <user> (string) - Nome do usuário

Descrição: Retorna as equipes de Pokémon associadas a um usuário específico.

Criar uma nova equipe de Pokémon

**Método: POST**

Rota: */api/teams*

Corpo da Solicitação: JSON com os campos user (string) e team (lista de strings)

Descrição: Cria uma nova equipe de Pokémon associada a um usuário fornecido, verificando a existência dos Pokémon na PokeAPI.

**Arquivos**

models/teams.json: Arquivo JSON utilizado para armazenar as equipes de Pokémon.

**Estrutura do Arquivo JSON**

O arquivo JSON segue a seguinte estrutura:

{

"user": "sleao",

"team": [

"blastoise",

"pikachu",

"charizard",

"venusaur",

"lapras",

"dragonite"

]

}