

Microarquitetura Back - End

A microarquitetura apresentada é uma aplicação back-end construída usando o framework Flask. A aplicação gerencia tarefas e jornadas do herói, permitindo a criação, atualização, exclusão e recuperação desses dados por meio de uma API RESTful.

Classes Básicas (Models)

Task (teste)

- Atributos:
 - **id**: Identificador único da tarefa.
 - **nome**: Nome da tarefa
 - **descrição**: Descrição da tarefa.
 - **status**: Status da tarefa (disponíveis, emAndamento ou concluidas).

Jornada

- Atributos:
 - **nome**: Nome único da jornada.
 - **tarefas**: Lista de tarefas associadas à jornada.

Banco de Dados (db.json)

`load_db()`

- Descrição: Carrega as tarefas e jornadas armazenadas no arquivo **db.json** ou inicializa um conjunto vazio se o arquivo não existir.
- Retorno: Dicionário contendo as tarefas e jornadas.

`save_db(tasks)`

- Descrição: Salva as tarefas e jornadas no arquivo **db.json**
- Parâmetros
 - **tasks**: Dicionário contendo as tarefas e jornadas a serem salvas.

`validate_jornada_payload(payload)`

- Descrição: Verifica se um payload de jornada contém pelo menos uma tarefa.
- Parâmetros:
 - **payload**: Dados da jornada a serem validados.

`initialize_tasks()`

- Descrição: Inicializa as tarefas, carregando-as do banco de dados ou criando um conjunto vazio se não existirem.
- Retorno: Dicionário contendo as tarefas.

Rotas para Jornadas do Herói (CRUD)

CREATE

- Rotas: `/task (POST)`, `/jornadas (POST)`
- Funções: `criar_task()`, `criar_jornada()`
- Descrição: Criação de uma nova tarefa ou jornada.
- Retorno (`/task`): JSON indicando o sucesso da criação e o ID da tarefa.
- Retorno (`/jornadas`): JSON contendo todas as jornadas.

READ

- Rotas: `/task (GET)`, `/jornadas (GET)`, `/jornadas/<nome_jornada> (GET)`
- Funções: `get_tasks()`, `get_jornadas()`, `get_jornada()`
- Descrição: Recuperação de informações sobre tarefas e jornadas.
- Retorno (`/task`): JSON indicando o sucesso da criação e o ID da tarefa
- Retorno (`/jornadas`): JSON indicando o sucesso da criação e o nome da jornada.

UPDATE

- Rotas: `/task/<id> (PUT)`, `/jornadas/<nome_jornada> (PUT)`
- Funções: `update_task()`, `update_jornada()`
- Descrição: Atualização de informações sobre tarefas ou jornadas.
- Retorno (`/task`): JSON indicando o sucesso da atualização e os detalhes da tarefa atualizada.
- Retorno (`/jornadas`): JSON indicando o sucesso da atualização e as tarefas atualizadas.

DELETE

- Rotas: `/task/<id> (DELETE)`, `/jornadas/<nome_jornada> (DELETE)`
- Funções: `delete_task()`, `delete_jornada()`
- Descrição: Exclusão de tarefas ou jornadas.
- Retorno (`/task`): Retorno: JSON indicando o sucesso da exclusão.
- Retorno (`/jornadas`): Retorno: JSON indicando o sucesso da exclusão.