

Manual de Instruções

A API de Personagens é uma API RESTful desenvolvida em Flask que permite a criação, leitura, atualização e exclusão de personagens fictícios. A API foi projetada para fornecer uma interface de programação simples e intuitiva para manipulação de personagens em um banco de dados. Este manual do usuário fornecerá uma visão geral do funcionamento básico dos endpoints disponíveis na API.

URL Base da API:

`http://localhost:5000`

Endpoints Disponíveis:

Consultar todos os personagens (GET):

Endpoint: `/personagens`

Método HTTP: GET

Descrição: Este endpoint retorna todos os personagens cadastrados no banco de dados.

Parâmetros de Consulta: Nenhum.

Resposta:

Código de Status 200 OK

Corpo da Resposta: Lista de objetos de personagem, cada um contendo os seguintes campos:

nome (string): Nome do personagem.

descricao (string): Descrição do personagem.

link_imagem (string): Link da imagem do personagem.

programa (string): Programa associado ao personagem.

animador (string): Animador responsável pelo personagem.

Consultar um personagem por ID (GET):

Endpoint: /personagens/{id}

Método HTTP: GET

Descrição: Este endpoint retorna os detalhes de um personagem específico com base em seu ID.

Parâmetros de Consulta:

id (integer): O ID do personagem desejado.

Resposta:

Código de Status 200 OK

Corpo da Resposta: Objeto de personagem contendo os seguintes campos:

nome (string): Nome do personagem.

descricao (string): Descrição do personagem.

link_imagem (string): Link da imagem do personagem.

programa (string): Programa associado ao personagem.

animador (string): Animador responsável pelo personagem.

Criar um novo personagem (POST):

Endpoint: /personagens

Método HTTP: POST

Descrição: Este endpoint permite criar um novo personagem no banco de dados.

Parâmetros de Entrada:

nome (string): Nome do personagem (obrigatório).

descricao (string): Descrição do personagem (obrigatório).

link_imagem (string): Link da imagem do personagem (obrigatório).

programa (string): Programa associado ao personagem (obrigatório).

animador (string): Animador responsável pelo personagem (obrigatório).

Resposta:

Código de Status 200 OK

Corpo da Resposta: Objeto de personagem criado, contendo os seguintes campos:

nome (string): Nome do personagem.

descricao (string): Descrição do personagem.

link_imagem (string): Link da imagem do personagem.

programa (string): Programa associado ao personagem.

animador (string): Animador responsável pelo personagem.

Atualizar um personagem por ID (PUT):

Endpoint: /personagens/{id}

Método HTTP: PUT

Descrição: Este endpoint permite atualizar os detalhes de um personagem específico com base em seu ID.

Parâmetros de Entrada:

id (integer): O ID do personagem a ser atualizado (obrigatório).

nome (string): Nome do personagem.

descricao (string): Descrição do personagem.

link_imagem (string): Link da imagem do personagem.

programa (string): Programa associado ao personagem.

animador (string): Animador responsável pelo personagem.

Resposta:

Código de Status 200 OK

Corpo da Resposta: Objeto de personagem atualizado, contendo os seguintes campos:

nome (string): Nome do personagem.

descricao (string): Descrição do personagem.

link_imagem (string): Link da imagem do personagem.

programa (string): Programa associado ao personagem.

animador (string): Animador responsável pelo personagem.

Excluir um personagem por ID (DELETE):

Endpoint: /personagens/{id}

Método HTTP: DELETE

Descrição: Este endpoint permite excluir um personagem específico com base em seu ID.

Parâmetros de Consulta:

id (integer): O ID do personagem a ser excluído.

Resposta:

Código de Status 204 No Content

Corpo da Resposta: Nenhum.

Autenticação:

A API de Personagens não requer autenticação para uso dos endpoints disponíveis. Qualquer usuário pode consultar, criar, atualizar ou excluir personagens.

Exemplos de Uso:

Aqui estão alguns exemplos de uso da API de Personagens utilizando o cURL, um popular utilitário de linha de comando para fazer requisições HTTP:

1.Consultar todos os personagens:

```
curl -X GET http://localhost:5000/personagens
```

2.Consultar um personagem por ID:

```
curl -X GET http://localhost:5000/personagens/{id}
```

3.Criar um novo personagem:

```
curl -X POST http://localhost:5000/personagens -d  
'nome=NomeDoPersonagem&descricao=DescricaoDoPersonagem&link_imagem=LinkDaImagem&pr  
ograma=ProgramaAssociado&animador=AnimadorResponsavel'
```

4.Atualizar um personagem por ID:

```
curl -X PUT http://localhost:5000/personagens/{id} -d  
'nome=NovoNomeDoPersonagem&descricao=NovaDescricaoDoPersonagem&link_imagem=NovoLin  
kDaImagem&programa=NovoProgramaAssociado&animador=NovoAnimadorResponsavel'
```

5.Excluir um personagem por ID:

```
curl -X DELETE http://localhost:5000/personagens/{id}
```

Nota: Substitua {id} pelos valores reais do ID do personagem desejado nos exemplos acima.

Espero que este manual do usuário seja útil para entender o funcionamento básico dos endpoints disponíveis na API de Personagens e como utilizá-los corretamente para manipular personagens fictícios em um banco de dados. Caso tenha alguma dúvida ou precise de mais informações, consulte a documentação completa da API no repositório github