

## API-ANIMESAO

### EM DESENVOLVIMENTO

#### User

1. **POST /user:** Cria um novo usuário com base nos detalhes fornecidos no corpo da requisição.

```
{
  "username": "joao123",
  "email": "joao@example.com",
  "password": "senha123",
  "imgUrl": "https://example.com/avatar.jpg"
}
```

2. **GET /user:** Retorna uma lista de todos os usuários cadastrados.
3. **PATCH /user/:id:** Atualiza os detalhes de um usuário específico com base no ID fornecido na URL e nos detalhes atualizados fornecidos no corpo da requisição.
4. **GET /user/:id:** Retorna os detalhes de um usuário específico com base no ID fornecido na URL.
5. **DELETE /user/:id:** Exclui um usuário específico com base no ID fornecido na URL.
6. **POST /user/login:** Autentica um usuário com base nas credenciais fornecidas no corpo da requisição e gera um token de acesso. Se as credenciais forem válidas, retorna o usuário autenticado junto com o token de acesso. Se as credenciais forem inválidas, lança uma exceção de não autorizado.

```
{
  "username": "joao123",
  "password": "senha123"
}
```

#### Rating

1. **POST /rating/:animeId/:userId:** Cria uma nova classificação para um anime específico pelo usuário especificado.

```
{
  "rating": 4
}
```

2. **PUT /rating/:id:** Atualiza os detalhes de uma classificação existente com base no ID fornecido.
3. **GET /rating/:id:** Retorna os detalhes de uma classificação específica com base no ID fornecido.

4. **DELETE /rating/:id**: Exclui uma classificação específica com base no ID fornecido.
5. **GET /rating**: Calcula o ranking de todos os animes com base nas classificações dos usuários e retorna uma lista de animes ordenada pelo ranking. OBS: FAZ O RANK DE TODOS OS ANIMES COM BASE NA SUAS CLASSIFICACOES FAZENDO ASSIM UMA MÉDIA
6. **GET /rating/:idAnime/:idUser**: Retorna a classificação de um anime específico feita por um usuário específico com base nos IDs fornecidos.

#### Historic

1. **GET /historic**: Retorna uma lista de todo o histórico registrado.
2. **GET /historic/:id**: Retorna os detalhes de um registro específico no histórico com base no ID fornecido.
3. **POST /historic**: Cria um novo registro no histórico com base no ID do usuário e no ID do episódio fornecidos no corpo da requisição.

```
{
  "userId": 1,
  "episodeId": 5
}
```

4. **PUT /historic/:id**: Atualiza os detalhes de um registro específico no histórico com base no ID fornecido.
5. **DELETE /historic/:id**: Exclui um registro específico no histórico com base no ID fornecido. Retorna o resultado da exclusão.

#### Genre(genero de anime)

1. **GET /genre**: Retorna uma lista de todos os gêneros.
2. **GET /genre/:id**: Retorna os detalhes de um gênero específico com base no ID fornecido.
3. **POST /genre**: Cria um novo gênero com base nos detalhes fornecidos no corpo da requisição.

```
{
  "genero": "Ação"
}
```

4. **PUT /genre/:id**: Atualiza os detalhes de um gênero específico com base no ID fornecido e nos detalhes atualizados fornecidos no corpo da requisição.
5. **DELETE /genre/:id**: Exclui um gênero específico com base no ID fornecido. Retorna o resultado da exclusão.

## Favorite

1. **GET /favorite:** Retorna uma lista de todos os favoritos registrados.
2. **GET /favorite/:id:** Retorna os detalhes de um favorito específico com base no ID fornecido.
3. **POST /favorite:** Cria um novo favorito com base no ID do anime e no ID do usuário fornecidos no corpo da requisição.

```
{
  "userId": 1,
  "animeId": 5
}
```

4. **PUT /favorite/:id:** Atualiza os detalhes de um favorito específico com base no ID fornecido e nos detalhes atualizados fornecidos no corpo da requisição.
5. **DELETE /favorite/:id:** Exclui um favorito específico com base no ID fornecido. Retorna o resultado da exclusão.

## Episode

1. **GET /episode:** Retorna uma lista de todos os episódios registrados.
2. **GET /episode/:id:** Retorna os detalhes de um episódio específico com base no ID fornecido.
3. **POST /episode:** Cria um novo episódio com base no ID do anime fornecido no corpo da requisição.

```
{
  "animeId": 1,
  "numeroEpisodio": 1,
  "tituloEpisodio": "Título do Episódio",
  "sinopseEpisodio": "Sinopse do Episódio",
  "urlVideo": "http://exemplo.com/video1",
  "servidorHospedagem": "Servidor de Hospedagem"
}
```

4. **PUT /episode/:id:** Atualiza os detalhes de um episódio específico com base no ID fornecido e nos detalhes atualizados fornecidos no corpo da requisição.
5. **DELETE /episode/:id:** Exclui um episódio específico com base no ID fornecido. Retorna o resultado da exclusão.

## Comments

1. **GET /comments:** Retorna uma lista de todos os comentários registrados.
2. **POST /comments:** Cria um novo comentário com base nos dados fornecidos no corpo da requisição, que incluem o usuário associado ao comentário e o ID do episódio ao qual o comentário se refere.

```
{  
  
  "content": "Conteúdo do comentário",  
  
  "userId": 1,  
  
  "episodeId": 5  
  
}
```

3. **PUT /comments/:id:** Atualiza os detalhes de um comentário específico com base no ID fornecido e nos detalhes atualizados fornecidos no corpo da requisição.
4. **GET /comments/:id:** Retorna os detalhes de um comentário específico com base no ID fornecido.
5. **DELETE /comments/:id:** Exclui um comentário específico com base no ID fornecido.

## AnimesType(Genre anime)

1. **GET /anime-type:** Retorna uma lista de todos os tipos de anime registrados.
2. **POST /anime-type:** Cria um novo tipo de anime com base nos dados fornecidos no corpo da requisição, que incluem o ID do anime e o ID do gênero associado.

```
{  
  "animeId": 1,  
  "generoId": 3  
}
```

3. **PUT /anime-type/:id:** Atualiza os detalhes de um tipo de anime específico com base no ID fornecido e nos detalhes atualizados fornecidos no corpo da requisição. Se o tipo de anime não for encontrado, lança uma exceção de "Tipo de anime não encontrado".
4. **GET /anime-type/:id:** Retorna os detalhes de um tipo de anime específico com base no ID fornecido. Se o tipo de anime não for encontrado, lança uma exceção de "Tipo de anime não encontrado".
5. **DELETE /anime-type/:id:** Exclui um tipo de anime específico com base no ID fornecido.
6. **GET /anime-type/byGenre/:genreId:** Retorna uma lista de animes associados a um gênero específico com base no ID do gênero fornecido. Filtro PESQUISA POR GENERO ANIME

`http://localhost:3000/anime-type/byGenre/:genreId`

## Anime

1. **GET /anime:** Retorna uma lista de todos os animes registrados.
2. **GET /anime/:id:** Retorna os detalhes de um anime específico com base no ID fornecido.
3. **POST /anime:** Cria um novo anime com base nos detalhes fornecidos no corpo da requisição.

```
{
  "titulo": "Título do Anime",
  "descricao": "Descrição do Anime",
  "imagemUrl": "http://exemplo.com/imagem.jpg",
  "dataLancamento": "2024-03-20",
  "diretor": "Nome do Diretor",
  "estudio": "Nome do Estúdio",
  "status": "Em Andamento",
  "numeroEpisodios": 12,
}
```

4. **PUT /anime/:id:** Atualiza os detalhes de um anime específico com base no ID fornecido e nos detalhes atualizados fornecidos no corpo da requisição.
5. **DELETE /anime/:id:** Exclui um anime específico com base no ID fornecido.
6. **GET /anime/search/:name:** Retorna uma lista de animes que correspondem ao nome fornecido. PESQUISA POR NOME DO ANIME:

`http://localhost:3000/anime/search/:name`

## Admin

1. **POST /admin:** Cria um novo administrador.

```
{
  "username": "nome_de_usuario",
  "password": "senha_segura",
  "name": "Nome do Administrador"
}
```

2. **GET /admin:** Retorna todos os administradores.
3. **GET /admin/:id:** Retorna um administrador específico.
4. **PATCH /admin/:id:** Atualiza parcialmente um administrador.
5. **DELETE /admin/:id:** Exclui um administrador.

6. **POST /admin/users/:userId**: Altera o status de um usuário. ATIVAR E DESTIVAR CONTA

`http://localhost:3000/admin/users/:userId`

7. **POST /admin/login**: Autentica um administrador.

```
{  
  "username": "nome_de_usuario",  
  "password": "senha_segura"  
}
```

## ATT

Vou adicionar novas entidades e novas rotas nas entidades já existente, vou atualizar esse ficheiro.