

soccer-stats-API

Endpoint para Login no Sistema

- **Método:** POST
- **URL:** `/login`
- **Descrição:** Autentica um usuário no sistema.
- **Corpo da Requisição (JSON):** `{ "username": "string", "password": "string" }`
- **Resposta de Sucesso (200):** Retorna um token JWT válido por 1 hora.

```
jsonCopy code
{ "token": "string" }
```

- **Resposta de Erro (401):** Retorna um erro se as credenciais forem inválidas ou o usuário não existir.

```
jsonCopy code
{ "error": "string" }
```

Endpoint para Listar Todas as Partidas

- **Método:** GET
- **URL:** `/matches`
- **Descrição:** Retorna uma lista com todas as partidas.
- **Resposta de Sucesso (200):** Retorna um array com os dados das partidas.

```
jsonCopy code
[
  { "id": "string", ... },
  { "id": "string", ... },
  ...
]
```

- **Resposta de Erro (500):** Retorna um erro interno do servidor caso ocorra algum problema na consulta.

Endpoint para Adicionar uma Nova Partida

- **Método:** POST
- **URL:** `/matches`
- **Descrição:** Adiciona uma nova partida ao sistema.
- **Corpo da Requisição (JSON):** Dados da nova partida.

```
jsonCopy code
{ "data": "string", ... }
```

- **Resposta de Sucesso (201):** Retorna uma mensagem indicando que a partida foi adicionada com sucesso.

```
jsonCopy code
{ "message": "string" }
```

- **Resposta de Erro (500):** Retorna um erro interno do servidor caso ocorra algum problema na adição da partida.

Endpoint para Atualizar Informações de Jogadores

- **Método:** PUT
- **URL:** `/players/:id`
- **Descrição:** Atualiza as informações de um jogador pelo seu ID.
- **Parâmetros da URL:** `id` - ID do jogador a ser atualizado.
- **Corpo da Requisição (JSON):** Dados atualizados do jogador.

```
jsonCopy code
{ "nome": "string", "posicao": "string", ... }
```

- **Resposta de Sucesso (200):** Retorna uma mensagem indicando que as informações foram atualizadas com sucesso.

```
jsonCopy code
{ "message": "string" }
```

- **Resposta de Erro (500):** Retorna um erro interno do servidor caso ocorra algum problema na atualização das informações.

Endpoint para Deletar um Time da Base de Dados

- **Método:** DELETE
- **URL:** `/teams/:id`
- **Descrição:** Deleta um time da base de dados pelo seu ID.
- **Parâmetros da URL:** `id` - ID do time a ser deletado.
- **Resposta de Sucesso (200):** Retorna uma mensagem indicando que o time foi deletado com sucesso.

```
jsonCopy code
{ "message": "string" }
```

- **Resposta de Erro (500):** Retorna um erro interno do servidor caso ocorra algum problema na exclusão do time.

Endpoint para Atualizar o lema de um Time

- **Método:** PUT
- **URL:** `/teams/:id/motto`
- **Descrição:** Atualiza o lema de um time pelo seu ID.
- **Parâmetros da URL:** `id` - ID do time cujo lema será atualizado.
- **Corpo da Requisição (JSON):** Novo lema do time.

```
jsonCopy code
{ "logo": "string" }
```

- **Resposta de Sucesso (200):** Retorna uma mensagem indicando que o lema do time foi atualizado com sucesso.

```
jsonCopy code
{ "message": "string" }
```

- **Resposta de Erro (500):** Retorna um erro interno do servidor caso ocorra algum problema na atualização do lema.