

Documentação GymClientControlApi

Por: Jhonatas R Correa

Visão Geral

A GymClientControlApi é uma API desenvolvida para o controle de clientes de uma academia. Ela oferece uma série de recursos para gerenciar eficientemente os clientes, conectando-se a um banco de dados SQL Server. A API utiliza protocolos HTTP e JSON para permitir a interação de forma simples e concisa com a aplicação.

Sobre suas funcionalidades, a API oferece os seguintes métodos HTTP:

- **GET**: Recupera todos os clientes cadastrados na academia que estão com o status ativo e os retorna em formato de lista.
- **GET/{document}**: Recupera o cliente correspondente ao documento fornecido como parâmetro.
- **POST**: Verifica se o documento inserido já existe na base de dados. Se não existir, realizar o cadastro do cliente com a matrícula ativada.
- **PUT/update/{document}**: Atualiza os dados do cadastro do cliente, sem permitir reativar a matrícula.
- **PUT/activate/{document}**: Reativa a matrícula do cliente na academia.
- **DELETE/{document}**: Realiza a exclusão lógica do cliente, desativando a matrícula, mas mantendo os dados registrados no banco.

As tecnologias, bibliotecas e métodos utilizados nesta API foram:

- Linguagem: C# 7.0
- Framework: .NET
- Conceitos: Arquitetura de Camadas e DDD (Domain-Driven Design)
- Práticas: Clean Code
- DTO's: Utilizados para transferência de dados entre as camadas da aplicação
- Testes: XUnit (framework de testes para .NET)
- Banco de Dados: SQL Server (Acessando via Dapper)
- IDE: Visual Studio 2022

Recursos e Endpoints

GET

Descrição

- Recupera todos os clientes cadastrados na academia que estão com o status ativo e os retorna em formato de lista.

Método HTTP

- **HTTP Get**

URL

- **URL Base:** *https://localhost:{porta}/api/v1/Clients*
- **URL EXEMPLO:** *https://localhost:44372/api/v1/Clients*

Parâmetros

- Não contém parâmetros.

Corpo da solicitação

- Não contém corpo de solicitação.

Exemplo de Json

- **Envio:** Não contém Json de envio na solicitação.
- **Retorno:**

Caso não haja nenhum cliente cadastrado e a conexão ocorra bem, o retorno manterá o 200, porém vazio. Se o base de dados já estiver com clientes registrados o retorno será uma lista de clientes ativos.

- **Exemplo de retorno:**

```
[
  {
    "name": "John Doe",
    "dateBirth": "1990-01-01T00:00:00",
    "phone": "+1 123-456-7890",
    "document": "1234567890",
    "email": "john.doe@example.com",
    "activeSubscription": true,
    "contractTime": 2,
    "registrationDate": "2023-07-16T01:58:30.95"
  },
  {
    "name": "marcos",
    "dateBirth": "2023-07-16T00:00:00",
    "phone": "987564321",
    "document": "15926",
    "email": "abnoim@hjaajda",
    "activeSubscription": true,
    "contractTime": 0,
    "registrationDate": "2023-07-16T03:36:35.183"
  }
]
```

Possíveis Retornos

- **HTTP 200 OK:** A solicitação foi bem-sucedida e os dados solicitados foram retornados com sucesso.

GET/{document}

Descrição

- Recupera o cliente correspondente ao documento fornecido como parâmetro.

Método HTTP

- **HTTP Get**

URL

- **Url Base:** *https://localhost:{porta}/api/v1/Clients/{documentoCliente}*
- **Url Exemplo:** *https://localhost:44372/api/v1/Clients/123*

Parâmetros

- [String] Documento do cliente o qual será realizado para encontrar o cliente e retorná-lo.

Corpo da solicitação

- Não contém corpo de solicitação.

Exemplo de Json

- **Envio:** Não contém Json de envio na solicitação.
- **Retorno:** Em caso de sucesso, o retorno será de apenas um cliente.

```
{  
  "name": "string",  
  "dateBirth": "2023-07-16T00:00:00",  
  "phone": "string",  
  "document": "123",  
  "email": "string",  
  "activeSubscription": true,  
  "contractTime": 1,  
  "registrationDate": "2023-07-16T03:07:10.75"  
}
```

Possíveis Retornos

- **HTTP 200 OK:** A solicitação foi bem-sucedida e os dados solicitados foram retornados com sucesso.
- **HTTP 404 Not Found:** Não foi encontrado algum cliente na base de dados com esse documento.

POST

Descrição

- O método verifica se o documento inserido já existe na base de dados. Se não existir, será realizado o cadastro do cliente com a matrícula ativada, caso exista, haverá um retorno indicando que o cliente com o respectivo documento já está cadastrado na base de dados.

Método HTTP

- **HTTP Post**

URL

- **Url Base:** *https://localhost:{porta}/api/v1/Clients*
- **Url Exemplo:** *https://localhost:44372/api/v1/Clients*

Parâmetros

- Não contém Parâmetros

Corpo da solicitação

- **Json de envio:**

```
{
  "name": "string",
  "document": "string",
  "contractTime": 0,
  "dateBirth": "2023-07-17T14:36:01.111Z",
  "email": "string",
  "phone": "string"
}
```

- **Campo a campo:**

[String] Name: Nome do respectivo cliente cadastrado.

[String] Document: Documento do respectivo cliente cadastrado (pensado a princípio como CPF, mas sem verificações referente a modelo e formatação)

[Enum] Contract Time: Enumerável referente ao tempo de matrícula fechado pelo aluno, seguindo a tabela a seguir.

0	Monthly (Mensal)
1	Quarterly (Trimestral)
2	Semester (Semestral)

[DateTime] DateBirth: Data de nascimento do cliente registrado

[String] Email: Email do cliente registrado

[String] Phone: Telefone do cliente registrado

Exemplo de Json

- **Envio:**

```
{
```

```

    "name": "string",
    "document": "string",
    "contractTime": 0,
    "dateBirth": "2023-07-17T14:36:01.111Z",
    "email": "string",
    "phone": "string"
  }

```

- **Retorno:**

```

{
  "document": "159481594815",
  "newClient": {
    "name": "string",
    "document": "159481594815",
    "contractTime": 0,
    "dateBirth": "2023-07-17T14:37:24.794Z",
    "email": "string",
    "phone": "string"
  }
}

```

Possíveis Retornos

- **HTTP 201 Created:** A solicitação de criação de um novo recurso foi bem-sucedida.
- **HTTP 400 Bad Request:** A solicitação foi malformada ou contém parâmetros inválidos. É importante verificar os detalhes da resposta para obter informações mais específicas sobre o problema encontrado.
- **HTTP 404 Not Found:** Não foi encontrado algum cliente na base de dados com esse documento.

PUT/update/{document}

Descrição

- Atualiza os **dados** do cadastro do cliente, porém, não reativa a matrícula do cliente.

Método HTTP

- **HTTP PUT**

URL

- **Url Base:** *https://localhost:{porta}/api/v1/Clients/update/{documentoDoCliente}*
- **Url Exemplo:** *https://localhost:44372/api/v1/Clients/update/123*

Parâmetros

- [String] Documento do cliente o qual será realizado para encontrar o cliente e efetuar a atualização dos dados.

Corpo da solicitação

- **Json de envio**

```
{
  "name": "string",
  "dateBirth": "2023-07-17T14:56:33.723Z",
  "phone": "string",
  "email": "string"
}
```

- **Campo a campo**

[string] Name: Nome do respectivo cliente cadastrado.

[DateTime] DateBirth: Data de nascimento do cliente registrado

[String] Phone: Telefone do cliente registrado

[String] Email: Email do cliente registrado

Exemplo de Json

- **Envio:**

```
{
  "name": "string",
  "dateBirth": "2023-07-17T14:56:33.723Z",
  "phone": "string",
  "email": "string"
}
```

- **Retorno:** Não contém Json de retorno.

Possíveis Retornos

- **HTTP 204 No Content:** A solicitação foi realizada com sucesso, porém não há retorno de algum objeto.
- **HTTP 404 Not Found:** Não foi encontrado algum cliente na base de dados com esse documento.

PUT/activate/{document}

Descrição

- Reativa a matrícula do cliente na academia.

Método HTTP

- **HTTP PUT**

URL

- **Url Base:** *https://localhost:{porta}/api/v1/Clients/activate/{documentoDoCliente}*
- **Url Exemplo:** *https://localhost:44372/api/v1/Clients/activate/123*

Parâmetros

- [String] Documento do cliente o qual será realizado para encontrar o cliente e efetuar a ativação do cadastro.

Corpo da solicitação

- Não contém corpo de solicitação .

Exemplo de Json

- **Envio:** Não contém Json de envio.
- **Retorno:** Não contém Json de retorno.

Possíveis Retornos

- **HTTP 204 No Content:** A solicitação foi realizada com sucesso, porém não há retorno de algum objeto.
- **HTTP 404 Not Found:** Não foi encontrado algum cliente na base de dados com esse documento.

DELETE/{document}

Descrição

- Realiza a exclusão lógica do cliente, desativando a matrícula, mas mantendo os dados registrados no banco.

Método HTTP

- **HTTP Delete**

URL

- **Url Base:** https://localhost:{porta}/api/v1/Clients/{DocumentoDoCliente}
- **Url Exemplo:** https://localhost:44372/api/v1/Clients/123

Parâmetros

- [String] Documento do cliente o qual será realizado para encontrar o cliente e efetuar a deleção lógica do seu registro (cliente).

Corpo da solicitação

- Não contém corpo de solicitação.

Exemplo de Json

- **Envio:** Não contém Json de Envio.
- **Retorno:** Não contém Json de Retorno.

Possíveis Retornos

- **HTTP 204 No Content:** A solicitação foi realizada com sucesso, porém não há retorno de algum objeto.
- **HTTP 404 Not Found:** Não foi encontrado algum cliente na base de dados com esse documento.