

Instituto Politécnico de Leiria

Escola Superior de Tecnologia e Gestão

2019/2020

Unidade Curricular  
**Serviços e Interoperabilidade de Sistemas (SIS)**

**Relatório de Implementação do Projeto**

Autores:

Gonçalo Manuel Ferreira de Oliveira Nº2180654

Ricardo Fernandes Marques Nº2180604

Samuel Jorge Perdigão Martins Nº2180641

Índice

[Índice 2](#_Toc29679130)

[Introdução 3](#_Toc29679131)

[Preparar a Base de Dados 4](#_Toc29679132)

[API REST 5](#_Toc29679133)

[Métodos (CRUD/CUSTOM) 5](#_Toc29679134)

[Testar com Postman 5](#_Toc29679135)

[Testar Requests com cURL 8](#_Toc29679136)

[PerfisController 9](#_Toc29679137)

[PlanosController 11](#_Toc29679138)

[ProdutosController 15](#_Toc29679139)

[Conclusão 16](#_Toc29679140)

Introdução

No âmbito de concluir o Curso TeSP de Programação de Sistemas de Informação, foi proposta realização de um projeto final de curso que engloba os temas lecionados nas Unidades Curriculares sob uma situação real deixada a critério dos estudantes.

Com isto, sugerimos a realização de um sistema que permite a gestão de uma cadeia de ginásios denominada de SportGym composto por um website e uma aplicação móvel, sendo a comunicação destes feita através de uma API REST.

Este relatório visa explicar aos docentes da Unidade Curricular de Serviços e Interoperabilidade de Sistemas (SIS) a implementação, funcionalidades, e, acima de tudo, como testar a API. Abaixo estão descritos os *endpoints*, como invocar cada método individualmente usando a linha de comandos com cURL e como testar as funcionalidades baseadas em *messaging.*

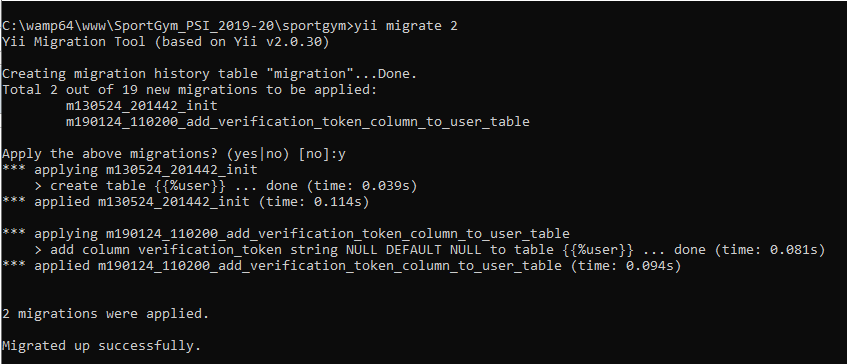
Preparar a Base de Dados

Passos:

1º Criar no phpMyAdmin a base de dados “sportgymdb”;

\*Na diretoria root do projeto\*

2º Executar o comando “yii migrate 2”;



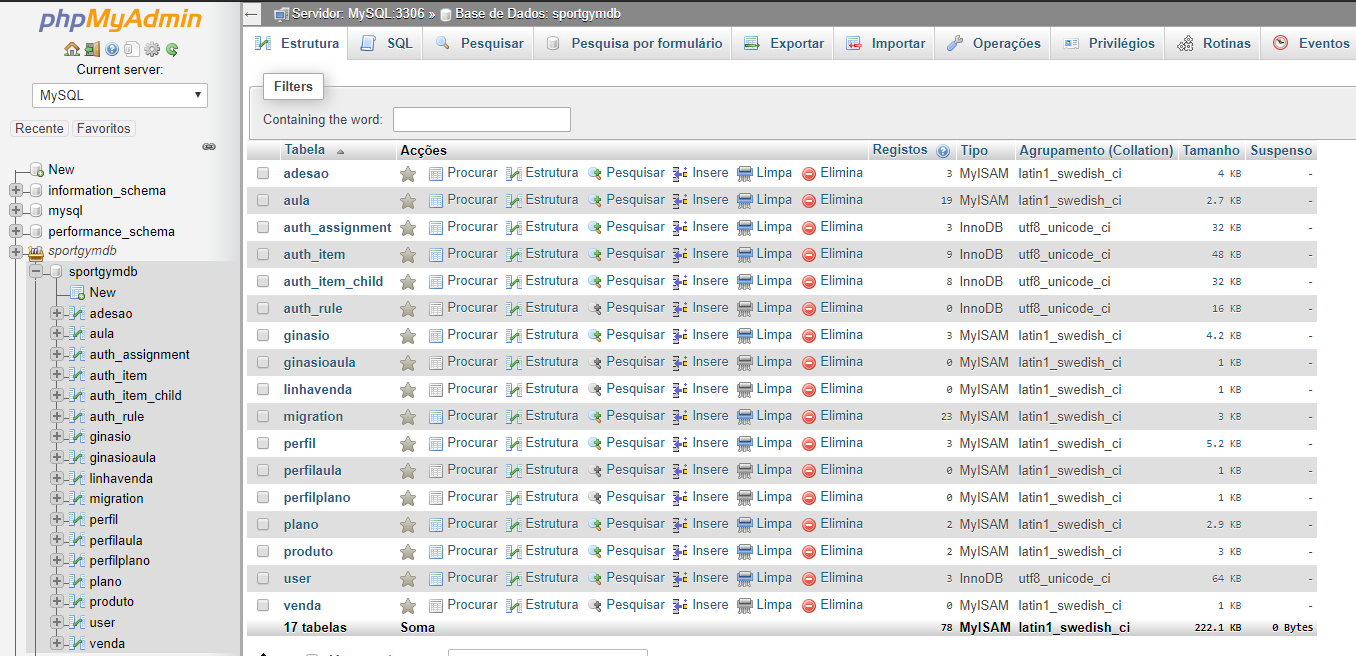
3º Executar o comando “yii migrate --migrationPath=@yii/rbac/migrations”;



4º Executar o comando “yii migrate”;



5º Deverá ter uma base de dados igual à da imagem abaixo.



API REST

# Métodos (CRUD/CUSTOM)

## Testar com Postman

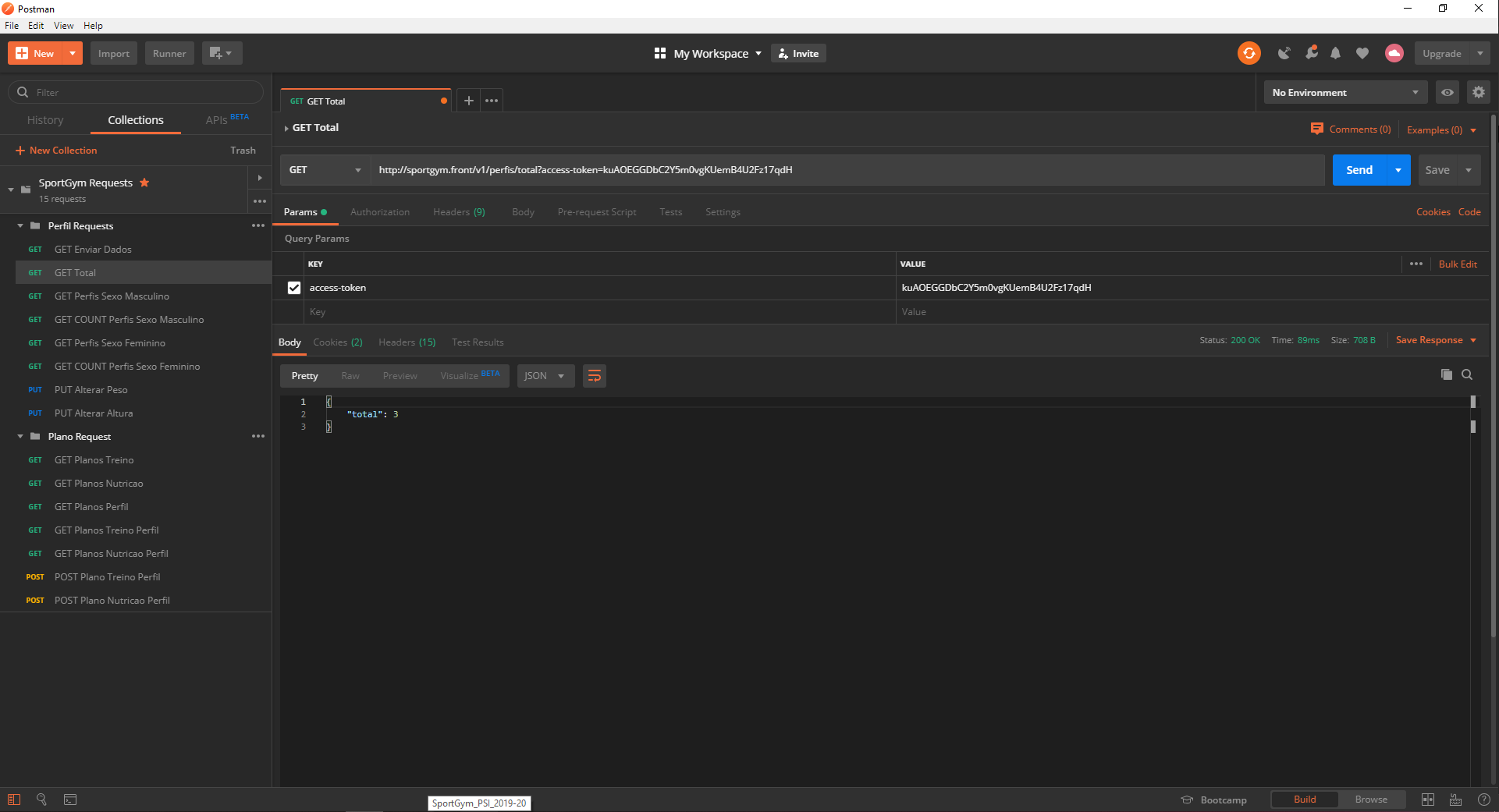
Tendo em conta o tema do projeto, foram desenvolvidos vários métodos, CRUD e CUSTOM, que permitem enviar diversos tipos

Para testar os *requests* dos métodos usou-se o Postman, que é um programa que facilita a realização de testes de integração utilizado a API. Desta forma também utilizámos este programa para a criação de uma *collection* que contém todos os URL’s necessários para testar ditos métodos.

Para testar os *requests* usando a *collection* basta importar, através do Postman, o ficheiro “SportGym Requests.postman\_collection.json” como mostra a figura abaixo.

Para que os *request* da *collection* funcionem corretamente, pode ir buscar o *access-token* do utilizador desejado no phpMyAdmin dentro da tabela “user” da base dados criada anteriormente.

1



2

5

4

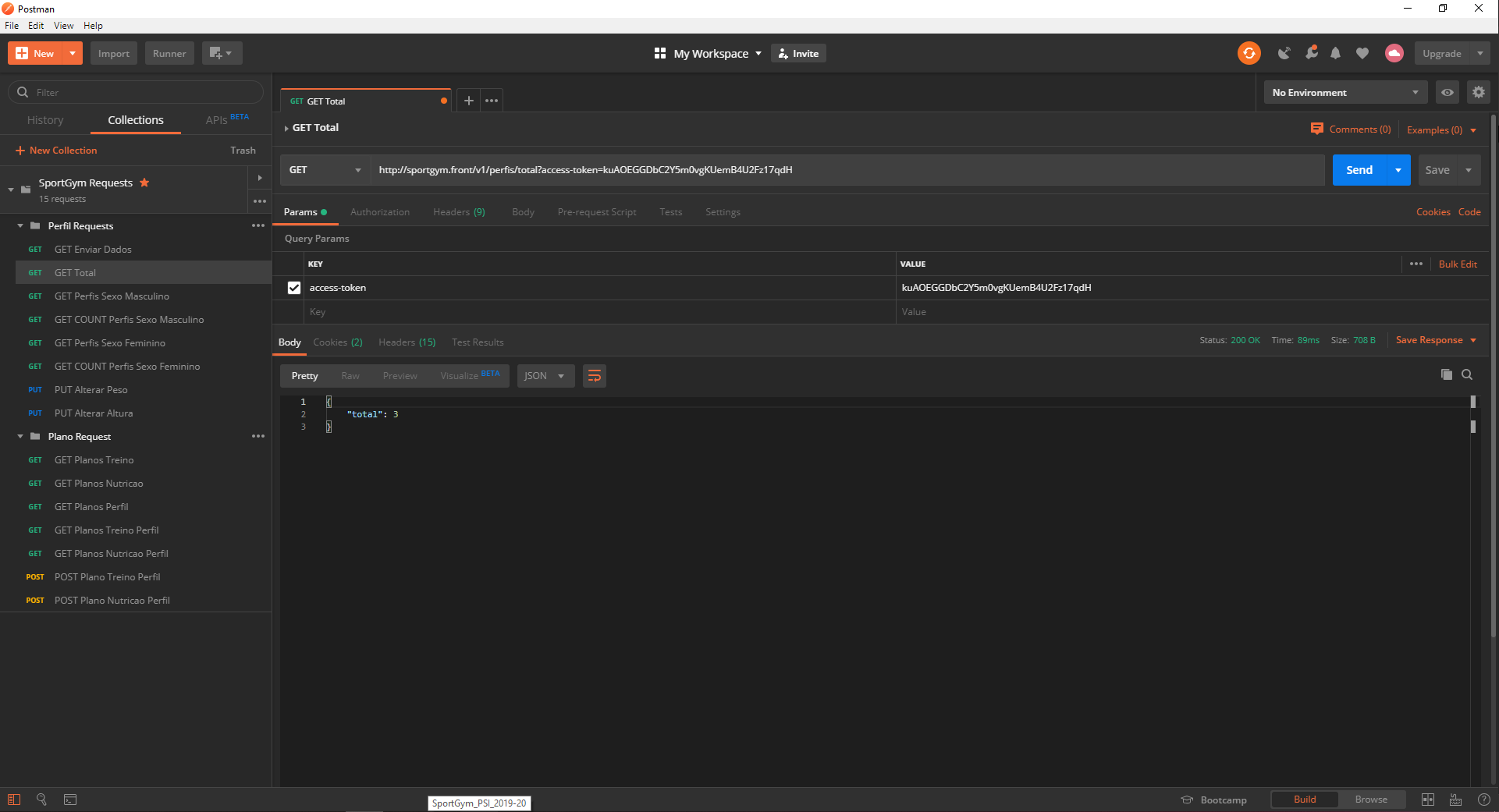
3

Figura : Layout do Postman

1. Permite importar ficheiros (usar para importar a *collection*);
2. Mostra todos os *requests* da *collection*;
3. Mostra o URL do *request* selecionado;
4. Envia *request* selecionado;
5. Mostra o *Output* do *request* e permite verificar se foi bem executado;

### Testar Requests do tipo GET com Postman

Para testar um *request* basta selecioná-lo da lista da *collection*, carregar no botão “Send” e observar o *Output* como apresentado na figura 2.



Output

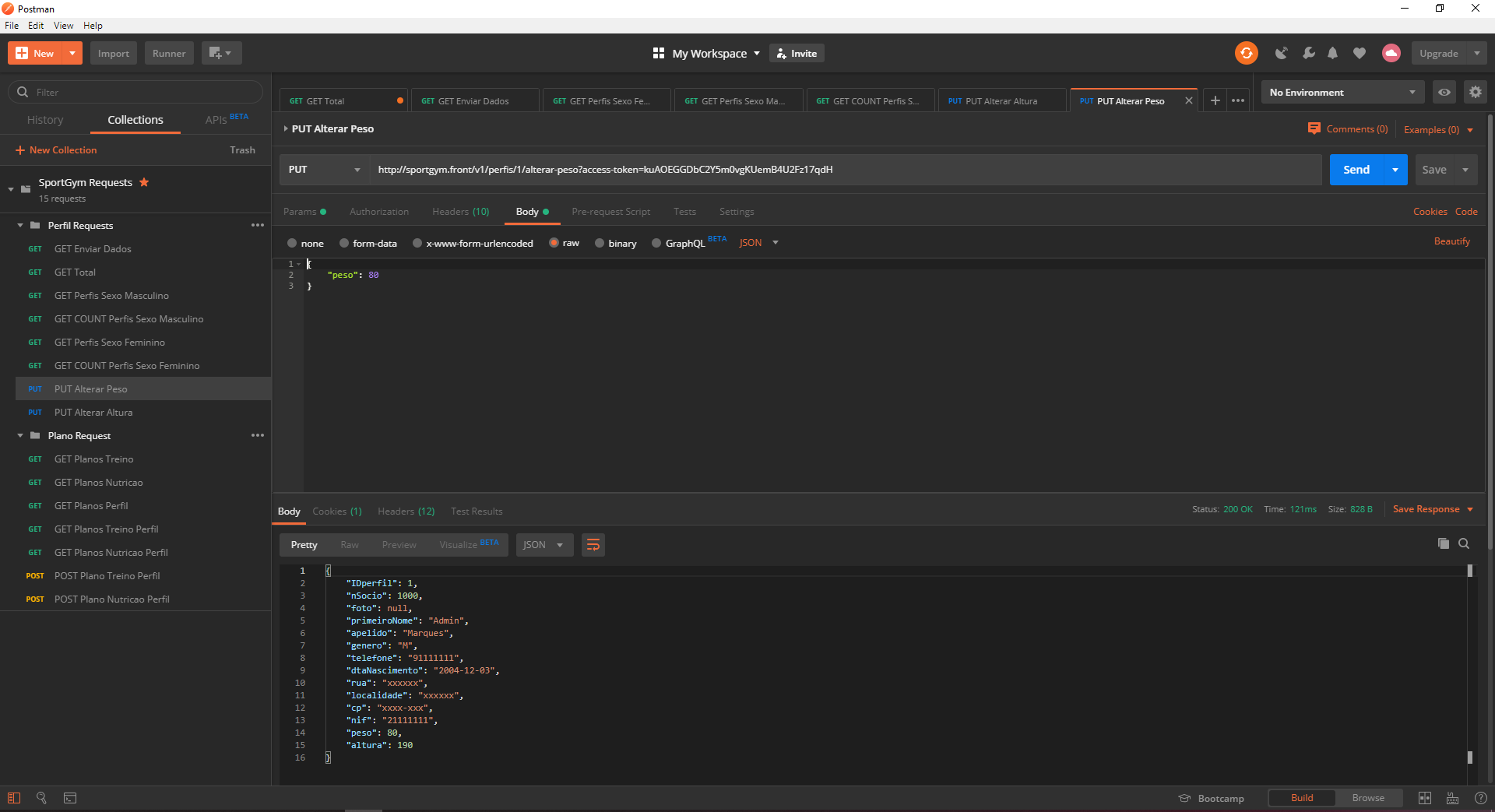
Envia o request

Lista dos requests

Figura

### Testar Requests do tipo PUT com Postman

Para testar um *request* do tipo PUT basta selecioná-lo da lista da *collection*, preencher no *body* campos a alterar, carregar no botão “Send” e observar o *Output* como apresentado na figura 3.



Indicar quais os campos a alterar

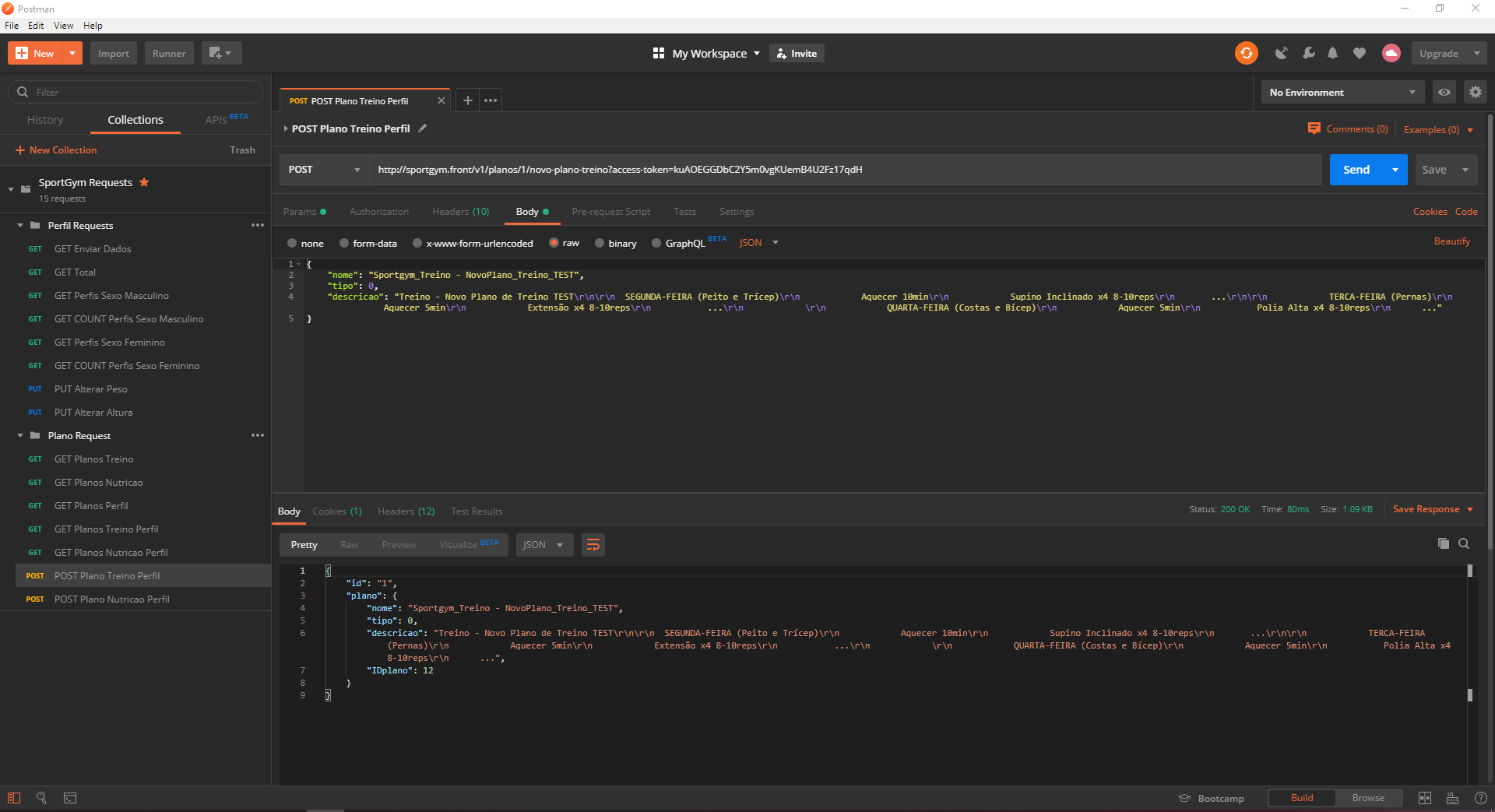
Output

Campo alterado

Figura

### Testar Requests do tipo POST com Postman

Para testar um *request* do tipo POST basta selecioná-lo da lista da *collection*, preencher no *body* campos do objeto a criar, carregar no botão “Send” e observar o *Output* como apresentado na figura 4.



Output

Indicar quais os campos a criar

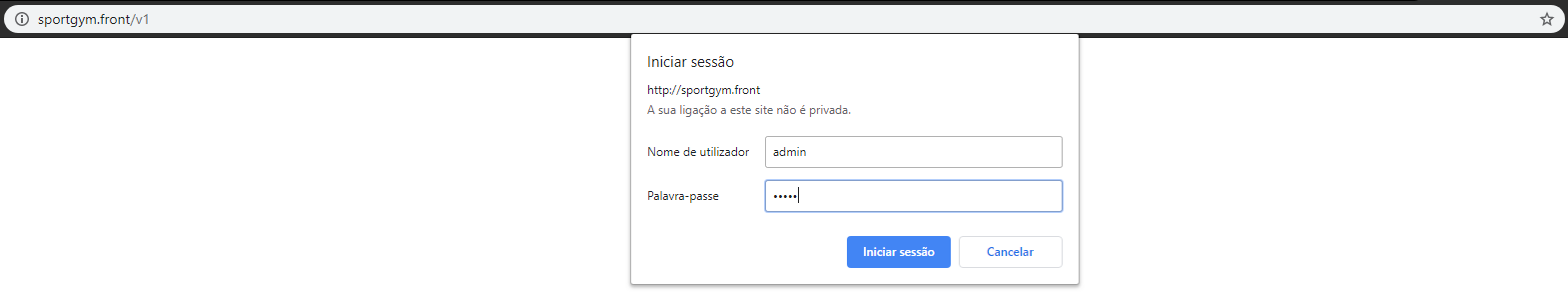
Figura

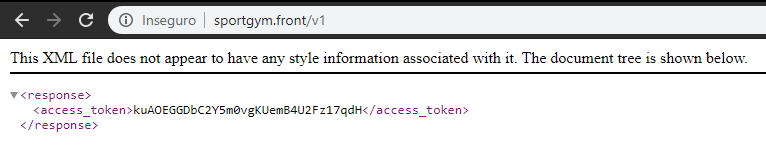
# Testar Requests com cURL

Como alternativa ao Postman, pode-se usar a linha de comandos com cURL e inserir os *Endpoints* de comunicação.

Abaixo estão listados os comandos a utilizar para testar os *requests* dos controladores bem como os *Outputs* esperados.

É importante salientar que para que os comandos funcionem conforme o esperado, é necessário ir buscar o access-token do utilizador logado. Isso pode ser feito ao aceder no browser o seguinte link: http://localhost/SportGym\_PSI\_2019-20/sportgym/frontend/web/v1. Ou então se tiver um virtualhost configurado no wamp64 pode usar o link: http://sportgym.front/v1

Após entrar no link, preencha os campos como na imagem.

Com o login feito pode copiar o token disposto na página como na figura abaixo.

Access-Token (substituir nos comandos cURL)

## PerfisController

### GET Enviar Dados

Executa a “actionEnviarDados” que devolve alguns dados selecionados de um dado perfil (id = 1).

curl --location --request GET “http://sportgym.front/v1/perfis/1/enviar-dados?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

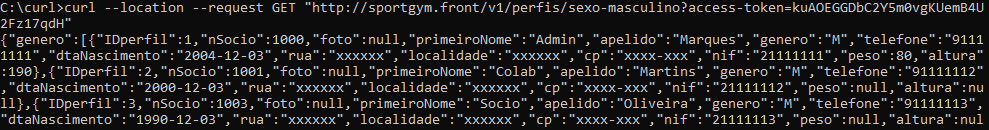
### GET Total

Executa a “actionTotal” que devolve o COUNT da quantidade de perfis existentes.

curl –location --request GET “http://sportgym.front/v1/perfis/total?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

### GET Pefis Sexo Masculino

Executa a “actionPerfisSexoMasculino” que mostra todos os perfis do género masculino.

curl --location --request GET “http://sportgym.front/v1/perfis/sexo-masculino?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

### GET COUNT Perfis Sexo Masculino

Executa a “actionTotalMasculino” que permite mostrar o COUNT do total de perfis do sexo masculino.

curl --location --request GET “http://sportgym.front/v1/perfis/total-masculino?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

### GET Perfis Sexo Feminino

Executa a “actionPerfisSexoFeminino” que mostra todos os perfis do género feminino.

curl --location --request GET “http://sportgym.front/v1/perfis/sexo-feminino?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”



### GET COUNT Perfis Sexo Feminino

Executa a “actionTotalFeminino” que mostra o COUNT do total de perfis do sexo feminino.

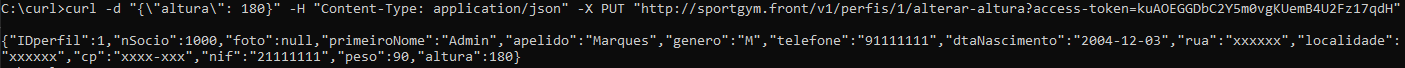
### PUT Alterar Peso

Executa a “actionAlterarPeso” que permite alterar peso de um certo perfil.

curl -d "{\"peso\": 80}" -H "Content-Type: application/json" -X PUT "http://sportgym.front/v1/perfis/1/alterar-peso?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN"

### PUT Alterar Altura

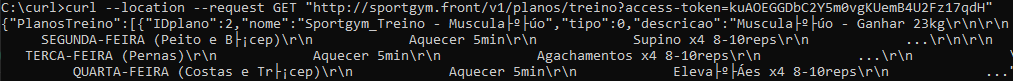
Executa a “actionAlterarAltura” que permite alterar peso de um certo perfil.

curl -d "{\"altura\": 180}" -H "Content-Type: application/json" -X PUT "http://sportgym.front/v1/perfis/1/alterar-altura?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN"

## PlanosController

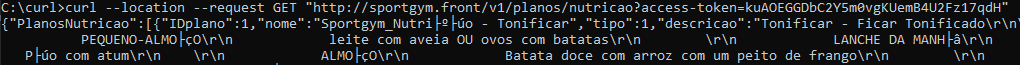
### GET Planos Treino

Executa a “actionPlanosTreino” que permite mostrar todos os planos do tipo treino.

curl --location --request GET “http://sportgym.front/v1/planos/treino?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

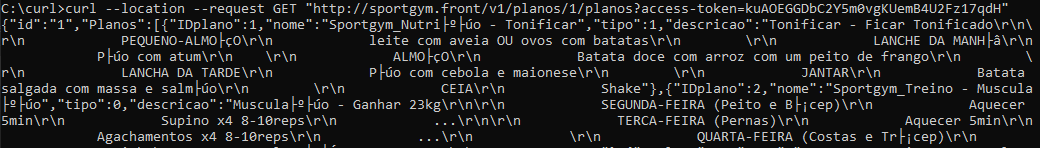
### GET Planos Nutricao

Executa a “actionPlanosNutricao” que permite mostrar todos os planos do tipo nutrição.

curl --location --request GET “http://sportgym.front/v1/planos/nutricao?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

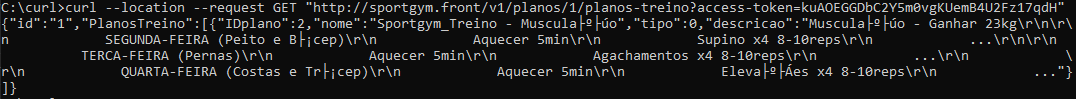
### GET Planos Perfil

Executa a “actionIdPlanos” que permite mostrar todos os planos atribuídos a um certo perfil.

curl --location --request GET “http://sportgym.front/v1/planos/1/planos?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN”

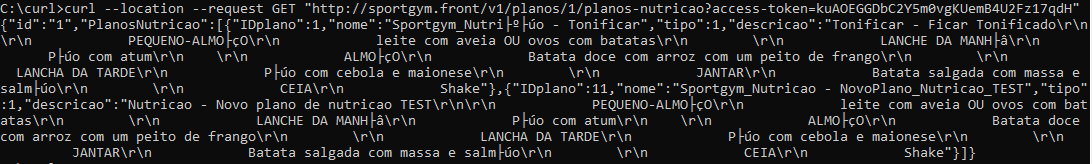
### GET Planos Treino Perfil

Executa a “actionIdIdPlanosTreino” que mostra todos os planos do tipo treino atribuídos a um certo perfil.

curl --location --request GET "http://sportgym.front/v1/planos/1/planos-treino?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN"

### GET Planos Nutricao Perfil

Executa a “actionIdIdPlanosNutricao” que mostra todos os planos do tipo nutrição atribuídos a um certo perfil.

curl --location –request GET "http://sportgym.front/v1/planos/1/planos-nutricao?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN"

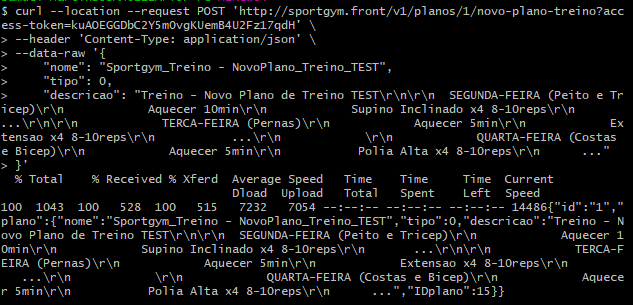
### POST Plano Treino Perfil

Executa a “actionIdCriarPlanoTreino” que, cria e atribuí a um perfil, um novo plano de treino. \*Nota: O comando abaixo é para ser executado usando o Git BASH\*

curl --location --request POST 'http://sportgym.front/v1/planos/1/novo-plano-treino?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN' \

--header 'Content-Type: application/json' \

--data-raw '{"nome": "Sportgym\_Treino - NovoPlano\_Treino\_TEST", "tipo": 0, "descricao": "Treino - Novo Plano de Treino TEST\r\n\r\n SEGUNDA-FEIRA (Peito e Tricep)\r\n Aquecer 10min\r\n Supino Inclinado x4 8-10reps\r\n ...\r\n\r\n TERCA-FEIRA (Pernas)\r\n Aquecer 5min\r\n Extensso x4 8-10reps\r\n ...\r\n \r\n QUARTA-FEIRA (Costas e Bicep)\r\n Aquecer 5min\r\n Polia Alta x4 8-10reps\r\n..."}'



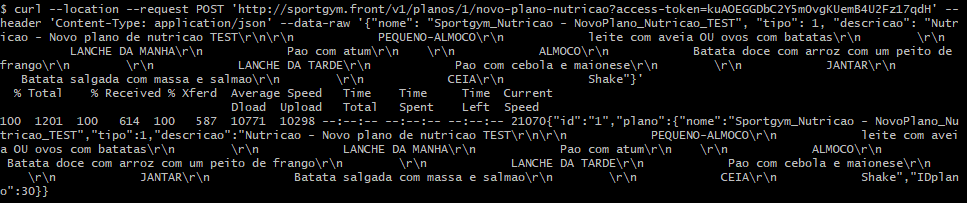
### POST Plano Nutricao Perfil

Executa a “actionIdCriarPlanoNutricao” que, cria e atribuí a um perfil, um novo plano de nutrição. \*Nota: O comando abaixo é para ser executado usando o Git BASH\*

curl --location --request POST 'http://sportgym.front/v1/planos/1/novo-plano-nutricao?access-token= INSERIR\_AQUI\_O\_ACCESS\_TOKEN' \

--header 'Content-Type: application/json' \

--data-raw '{"nome": "Sportgym\_Nutricao - NovoPlano\_Nutricao\_TEST", "tipo": 1, "descricao": "Nutricao - Novo plano de nutricao TEST\r\n\r\n PEQUENO-ALMOCO\r\n leite com aveia OU ovos com batatas\r\n \r\n LANCHE DA MANHA\r\n Pao com atum\r\n \r\n ALMOCO\r\n Batata doce com arroz com um peito de frango\r\n \r\n LANCHE DA TARDE\r\n Pao com cebola e maionese\r\n \r\n JANTAR\r\n Batata salgada com massa e salmao\r\n \r\n CEIA\r\n Shake"}'



## ProdutosController

### PUT Alterar Estado

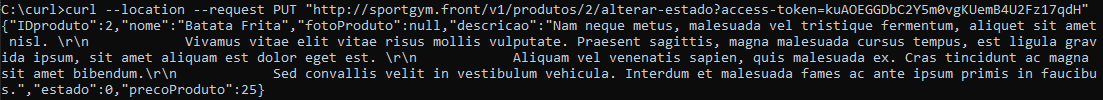
Executa a “actionAlterarEstado” que altera automaticamente o valor do estado do produto (0 ou 1). Também permite testar a autorização do utilizador autenticado, pelo que apenas o utilizador com estatuto de administrador, tem permissão para o fazer.

Para testar o acesso de um utilizador diferente, basta substituir no comando o acess-token para o de um utilizador diferente.

curl --location --request PUT "http://sportgym.front/v1/produtos/2/alterar-estado?access-token=INSERIR\_AQUI\_O\_ACCESS\_TOKEN\_DO\_ADMINISTRADOR"

Abaixo pode-se verificar o *output* caso o utilizador tenha permissão para tal.

curl --location --request PUT "http://sportgym.front/v1/produtos/2/alterar-estado?access-token= INSERIR\_AQUI\_O\_ACCESS\_TOKEN\_DO\_SOCIO"



*Output de um utilizador sem permissões*



Conclusão

Com o desenvolvimento desta API a nossa compreensão de como a comunicação e a partilha de dados entre dois sistemas de informação é realizada, foi alargada.

Porém a decisão da entrega deste projeto anteceder a dos restantes projetos não nos permitiu demonstrar o planeado, pelo que tivemos de recorrer apenas aos testes utilizando a linha de comandos.