V1.0 Integration Documentation

This document seeks to list the features contained in the integration API.

**Modification date: 22/05/2022**

**Authors:**

Nathan Ferreira Rodrigues Costa – Software Engineer

*Sumário*

[1. Environments 1](#_Toc71754076)

[2. Authentication: 2](#_Toc71754077)

[3. User 3](#_Toc71754078)

[3.1 Create User 4](#_Toc71754079)

[3.2 Update User 5](#_Toc71754080)

3.3 Delete User ................................................................................................................6

3.4 Find by id User……………………………………………………………………………………………………….0

3.5 Find all User …………………………………………………………………………………………………………..0

4. Person ……………………………………………………………………………………………………………………0

4.1 CreatePerson………………………………………………………………………………………………………….0

4.2 Update Person………………………………………………………………………………………………………0

4.3 Delete Person………………………………………………………………………………………………………..0

4.4 Find by id Person………………………………………………………………………………………………..0

4.5 Find all Person…………………………………………………………………………………………………….0

4.6 Create user for client……………………………………………………………………………………………0

5. Workouts…………………………………………………………………………………………………………….0

5.1 Create Workouts…………………………………………………………………………………………………0

5.2 Update Workouts………………………………………………………………………………………………..0

5.3 Delete Workouts………………………………………………………………………………………………….0

5.4 Find by id Workouts …………………………………………………………………………………………….0

5.5 Find all Workouts………………………………………………………………………………………………….0

5.6 Search all workouts by person……………………………………………………………………………...0

6 Comments……………………………………………………………………………………………………………..0

6.1 Create Comments………………………………………………………………………………………………….0

6.2 Update Comments …………………………………………………………………………………………….…0

6.3 Delete Comments ………………………………………………………………………………………………...0

6.4 Find by id Comments …………………………………………………………………………………………...0

6.5 Find all comments…………………………………………………………………………………………………0

6.6 Search all comments by workouts…………………………………………………………………………0

7.0 Check Workouts…………………………………………………………………………………….………………0

7.1 Create Check Workouts………………………………………………………………………………………0

7.2 Update Check Workouts ……………………………………………………………………………………..0

7.3 Delete Check Workouts ………………………………………………………………………………………0

7.4 Find by id Check Workouts …………………………………………………………………………………..0

7.5 Find all Check Workouts ……………………………………………………………………………………….0

7.6 Search all comments by workouts………………………………………………………………………..0

# Environments

dev: <https://localhost:80/>

prod: <https://nathan-dev.com/>

Github : <https://github.com/nathanfrc/gym-irlanda>

E-mail: [nathan.frc@hotmail.com](mailto:nathan.frc@hotmail.com)

Question contact

# Autentication:

**Método HTTP POST**

**{{base}}/login**

To carry out consultations on the services provided by the application, it will be

the authentication method described is required. Therefore, in the header of each

request must be informed the Header "Authorization" containing the word

“Bearer” followed by the generated token.

*Body*

*JSON*

Graphical user interface, text, application, email

Description automatically generated

*Response*

*HTTP Status 200*

*Http error 4XX, 5XX*

*Graphical user interface, text, application, email, website

Description automatically generated*

Graphical user interface, text, application

Description automatically generated

Link GitHub:

<https://github.com/nathanfrc/gym-irlanda>

# 3. User

## 3.1 Create user

**Method HTTP POST**

**{{base}}/user**

Description : create a new user

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Mandatory** | **Type** | **Description** |
| username | yes | String | Username of the login user |
| password | yes | String | Password of the login user |
| personId | yes | integer | id of the person who owns the login |
| roles | no | Array | Users permission |

*Types de roles permission:*

*ROLE\_ADMIN*

*ROLE\_STAFF*

*ROLE\_USER*

*Request / Response*

*Body*

*JSON*

Graphical user interface, text, application, email

Description automatically generated

## 3.2 Update user

**Method HTTP PUT**

**{{base}}/user/{{id}}** -> queryParam id user

Graphical user interface, text, application, email

Description automatically generated

## 3.3 Delete user

**Method HTTP PUT**

**{{base}}/user/{{id}}** -> queryParam id user

Graphical user interface, text, application

Description automatically generated

Response HTTP 204

## 3.4 find by id user

**Method HTTP GET**

**{{base}}/user/{{id}}** -> queryParam id user

search for a user by id

# Text Description automatically generated with low confidence

## 3.5 find all user

**Method HTTP GET**

**{{base}}/users**

search all users

A picture containing text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# 4. Person

## 4.1 create

**Method HTTP POST**

**{{base}}/person**

create a new person

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Mandatory** | **Type** | **Description** |
| firstName | yes | String | Person’s first name |
| lastName | yes | string | Person´s last name |
| numberId | yes | string | Unique identification number |
| email | yes | string | Person’s email |
| heigth | no | float | Person’s height |
| weigth | no | integer | Person’s weigth |

*Request*

*Json Body*

Graphical user interface, text, application

Description automatically generated

## 4.2 Update person

**Method HTTP PUT**

**{{base}}/person/{{id}}** -> queryParam id person

Graphical user interface, text, application

Description automatically generated

## 4.3 Delete person

**Method HTTP DELETE**

**{{base}}/person/{{id}}** -> queryParam id person

Graphical user interface, text, application

Description automatically generated

## 4.4 Find by id person

**Method HTTP GET**

**{{base}}/person/{{id}}** -> queryParam id person

Graphical user interface, text

Description automatically generated

## 4.5 Find all persons

**Method HTTP GET**

**{{base}}/persons**

Graphical user interface, text

Description automatically generated

## 4.6 Create user for client with client access permission only

**Method HTTP GET**

**{{base}}/person/user**

Graphical user interface, text

Description automatically generated

Table

Description automatically generated

# 5. Workouts

## 5.1 create workouts

**Method HTTP POST**

**{{base}}/workout**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Mandatory** | **Type** | **Description** |
| **title** | yes | String | **Training title** |
| description | yes | String | Training description for students |
| time | yes | Integer | Time the student will do the training |

Graphical user interface, text, application

Description automatically generated

## 5.2 update workouts

**Method HTTP PUT**

**{{base}}/workout/{{id}}** -> queryParam id workouts

Graphical user interface, text, application, email

Description automatically generated

## 5.3 delete workouts

**Method HTTP DELETE**

**{{base}}/workout/{{id}}** -> queryParam id workouts

Graphical user interface, text, application, email

Description automatically generated

## 5.4 find by id workouts

**Method HTTP GET**

**{{base}}/workout/{{id}}** -> queryParam id workouts

Graphical user interface, text

Description automatically generated

## 5.5 find all workouts

**Method HTTP GET**

**{{base}}/workouts**

Text

Description automatically generated

## 5.6 search all workouts by person

**Method HTTP GET**

**{{base}}/person/workouts/{id}** -> queryParam id workouts

Graphical user interface, text

Description automatically generated

Table

Description automatically generated

# 6. Comments

## 6.1 create comments

**Method HTTP POST**

**{{base}}/workout/comments**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Mandatory** | **Type** | **Description** |
| **description** | yes | String | **Comments entered** |
| starts | yes | Integer | Stars of enjoy the workout |
| workoutsId | yes | Integer | Workout id |

Graphical user interface, text, application

Description automatically generated

## 6.2 update comments

**Method HTTP PUT**

**{{base}}/workout/comments/{id}** -> queryParam id comments

Graphical user interface, text, application

Description automatically generated

## 6.3 delete comments

**Method HTTP DELETE**

**{{base}}/workout/comments/{id}** -> queryParam id comments

Graphical user interface, text, application, chat or text message

Description automatically generated

## 6.4 find by id

**Method HTTP GET**

**{{base}}/workout/comments/{id}** -> queryParam id comments

Graphical user interface, text

Description automatically generated

## 6.5 find all

**Method HTTP GET**

**{{base}}/workout/comments**

Graphical user interface, text

Description automatically generated

## 6.6 search all comments by workouts

**Method HTTP GET**

**{{base}}/workout/{id}/comments** -> queryParam id workouts

Text

Description automatically generated

# 7. Check Workouts

## 7.1 create check Workouts

**Method HTTP POST**

**{{base}}/checkworkout**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Mandatory** | **Type** | **Description** |
| workoutsId | yes | Integer | Workouts id |
| Training\_check | yes | Array | Days training |
|  |  |  |  |

Graphical user interface, application

Description automatically generated

## 7.2 update check Workouts

**Method HTTP PUT**

**{{base}}/checkworkout/{id} ->** -> queryParam id checworkouts

Graphical user interface, text, application

Description automatically generated

## 7.3 delete check Workouts

**Method HTTP DELETE**

**{{base}}/checkworkout/{id} ->** -> queryParam id checworkouts

Graphical user interface, text, application, email

Description automatically generated

## 7.4 find by id check Workouts

**Method HTTP GET**

**{{base}}/checkworkout/{id} ->** -> queryParam id checworkouts

Graphical user interface

Description automatically generated with medium confidence

## 7.5 find all check Workouts

**Method HTTP GET**

**{{base}}/checkworkout**

Graphical user interface, text, application

Description automatically generated

## 7.6 search all comments by workouts

**Method HTTP GET**

**{{base}}/checkworkout/person/{id} ->** -> queryParam id person

Graphical user interface, text

Description automatically generated