API Documentation

Referral Application Recruitment



Contents

[Document guide 4](#_Toc126599026)

[Version 4](#_Toc126599027)

[Authors 4](#_Toc126599028)

[API Testing 5](#_Toc126599029)

[Endpoints 6](#_Toc126599030)

[Systems 6](#_Toc126599031)

[Endpoint: <URL>/Systems 6](#_Toc126599032)

[/Systems/SeedDatabase 6](#_Toc126599033)

[/Systems/Unit test 6](#_Toc126599034)

[Authentication 7](#_Toc126599035)

[Endpoint: <URL>/Authentication 7](#_Toc126599036)

[POST: 7](#_Toc126599037)

[PUT: 7](#_Toc126599038)

[Users 8](#_Toc126599039)

[Endpoint: <URL>/Users 8](#_Toc126599040)

[GET: 8](#_Toc126599041)

[POST: 8](#_Toc126599042)

[PUT: 9](#_Toc126599043)

[DELETE: 9](#_Toc126599044)

[Profiles 10](#_Toc126599045)

[Endpoint: <URL>/Profiles 10](#_Toc126599046)

[GET: 10](#_Toc126599047)

[POST: 10](#_Toc126599048)

[PUT: 11](#_Toc126599049)

[DELETE: 11](#_Toc126599050)

[Tasks 12](#_Toc126599051)

[Endpoint: <URL>/Tasks 12](#_Toc126599052)

[GET: 12](#_Toc126599053)

[POST: 12](#_Toc126599054)

[PUT: 13](#_Toc126599055)

[DELETE: 13](#_Toc126599056)

[Scoreboards 14](#_Toc126599057)

[Endpoint: <URL>/Scoreboards 14](#_Toc126599058)

[GET: 14](#_Toc126599059)

[POST: 14](#_Toc126599060)

[PUT: 15](#_Toc126599061)

[DELETE: 15](#_Toc126599062)

[Rewards 16](#_Toc126599063)

[Endpoint: <URL>/Rewards 16](#_Toc126599064)

[GET: 16](#_Toc126599065)

[POST: 16](#_Toc126599066)

[PUT: 17](#_Toc126599067)

[DELETE: 17](#_Toc126599068)

[Referrals 18](#_Toc126599069)

[Endpoint: <URL>/Referrals 18](#_Toc126599070)

[GET: 18](#_Toc126599071)

[POST: 18](#_Toc126599072)

[PUT: 19](#_Toc126599073)

[DELETE: 19](#_Toc126599074)

[Candidates 20](#_Toc126599075)

[Endpoint: <URL>/Candidates 20](#_Toc126599076)

[GET: 20](#_Toc126599077)

[POST: 20](#_Toc126599078)

[PUT: 21](#_Toc126599079)

[DELETE: 21](#_Toc126599080)

# Document guide

This document is designed for developers who will use the Referral Application Recruitment API. In this document a description of each endpoint will be written. Each endpoint description contains a description of the usage of the endpoint, a table of fields and an example output.

The workflow of the application will be designed into the front-end application as the back-end only handles the provisioning, saving and deleting of data.

# Version

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | 28-11-2022 | Document creation |
| 1.1 | 16-01-2023 | Added remaining chapters |
| 1.2 | 06-02-2023 | Reviewed and updated for deliverable |

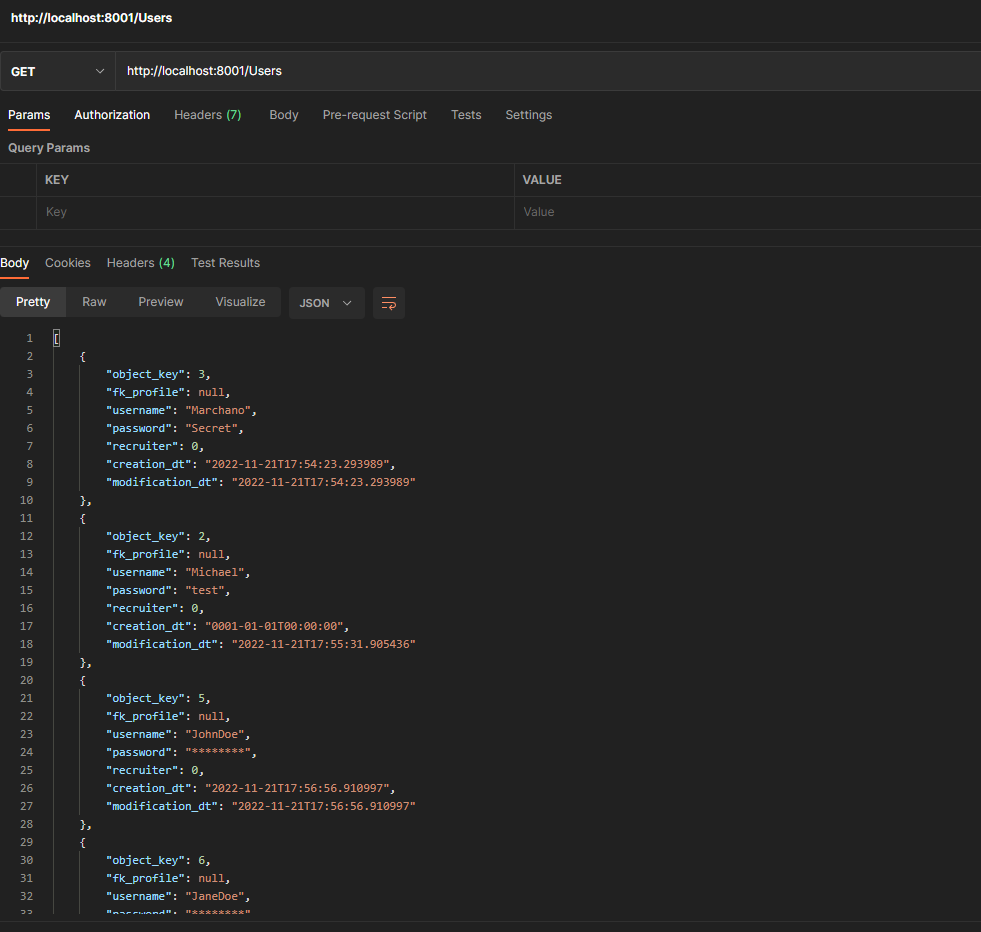
# Authors

|  |  |
| --- | --- |
| Name | Student number |
| Jeremey Sas | 1028294 |
| Michael Heerkens | 1028285 |
| Marchano Gopal | 1038165 |
| Tobias Roessingh | 1042623 |

# API Testing

The testing of the API is done with the Application named ‘Postman’, this application can be downloaded at: <https://www.postman.com/downloads/>.

The API testing handles the testing of Get requests, Post requests, Put requests, Delete requests and requests of which the API does not support the request-type, for example: Copy request.



# Endpoints

## Systems

### Endpoint: <URL>/Systems

### /Systems/SeedDatabase

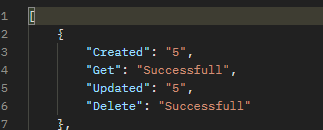
This endpoint will seed the database with an initial user with the recruiter privileges. The username + password combination for this user is:

* Username: admin
* Password: admin

### /Systems/Unit test

This endpoint will return a JSON list with the results of the unit tests, this is made for ease of testing.

Sample response:



## Authentication

### Endpoint: <URL>/Authentication

### POST:

The authentication POST request is used to log into the application.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| username | String | True | This is used to find an active user in the system |
| password | String | True | This will be used to authenticate the user |

### PUT:

The authentication PUT request is used to update the password of an existing user.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| fk\_user | Integer | True | This is used to find the user |
| password | String | True | This will be used to set the new password |
| password\_check | String | True | This will be used to compare the passwords and confirm the input is the same |

## Users

### Endpoint: <URL>/Users

### GET:

The users GET request will return a list of active system users. This list will be used to create, edit or delete users from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the user with this specific key |

### POST:

The users POST request is used to create new users into the active system. To successfully create a new user the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| username | String | True | This will be the login and display name of the user |
| password | String | True | This password will be used to log into the system if combined with the correct username |
| recruiter | Integer | True | This field should be either 0 or 1 depending on if this user should be a recruiter or an employee |
| fk\_profile | Integer | False | This is the object\_key property of a profile entity |
| profile | Object <Profile> | False | If the profile field is provided and filled with a valid profile object, the API will handle the creation of this new object and link It to the user being created |

### PUT:

The users PUT request is used to update an existing user of the active system. To successfully update an existing user the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing user |
| username | String | True | This will be the login and display name of the user |
| password | String | True | This password will be used to log into the system if combined with the correct username |
| recruiter | Integer | True | This field should be either 0 or 1 depending on if this user should be a recruiter or an employee |
| fk\_profile | Integer | False | This is the object\_key property of a profile entity |
| profile | Object <Profile> | False | If the profile field is provided and filled with a valid profile object, the API will handle the creation of this new object and link It to the user being created |

### DELETE:

The users DELETE request will delete an existing user from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the user entity with the provided object\_key value |

## Profiles

### Endpoint: <URL>/Profiles

### GET:

The profiles GET request will return a list of active profiles. This list can be used to create, edit or delete profiles from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the profile with this specific key |

### POST:

The profiles POST request is used to create new profiles into the active system. To successfully create a new profile the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| name | String | False | This is the first name for the profile entity |
| surname | String | False | This is the surname for the profile entity |
| email | String | False | This is the email for the profile entity |
| phone\_number | String | False | This is the phone number for the profile entity |
| address | String | False | This is the address for the profile entity |

### PUT:

The profiles PUT request is used to update an existing profile of the active system. To successfully update an existing profile the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing profile |
| name | String | False | This is the first name for the profile entity |
| surname | String | False | This is the surname for the profile entity |
| email | String | False | This is the email for the profile entity |
| phone\_number | String | False | This is the phone number for the profile entity |
| address | String | False | This is the address for the profile entity |

### DELETE:

The profiles DELETE request will delete an existing profile from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the profile entity with the provided object\_key value |

## Tasks

### Endpoint: <URL>/Tasks

### GET:

The tasks GET request will return a list of active tasks. This list can be used to create, edit or delete tasks from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the task with this specific key |

### POST:

The tasks POST request is used to create new tasks into the active system. To successfully create a new task the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| name | String | True | This is the name for the task entity |
| points | Integer | True | This is the amount of points a task can pay-out upon completion |
| description | String | False | This is the description of a task |

### PUT:

The tasks PUT request is used to update an existing task of the active system. To successfully update an existing task the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing task |
| name | String | False | This is the name for the task entity |
| points | Integer | False | This is the amount of points a task can pay-out upon completion |
| description | String | False | This is the description of a task |

### DELETE:

The tasks DELETE request will delete an existing task from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the task entity with the provided object\_key value |

## Scoreboards

### Endpoint: <URL>/Scoreboards

### GET:

The scoreboards GET request will return a list of active scoreboards. This list can be used to create, edit or delete scoreboards from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the task with this specific key |

### POST:

The scoreboards POST request is used to create new scoreboards into the active system. To successfully create a new scoreboard the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| fk\_user | Integer | True | This is should be the object\_key of the creator of the scoreboard |
| name | String | True | This is the name for the scoreboard entity |
| start\_dt | String | True | This is the start date for the scoreboard |
| end\_dt | String | True | This is the end date for the scoreboard |

### PUT:

The scoreboards PUT request is used to update an existing scoreboard of the active system. To successfully update an existing scoreboard the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing scoreboard |
| fk\_user | Integer | True | This is should be the object\_key of the creator of the scoreboard |
| name | String | True | This is the name for the scoreboard entity |
| start\_dt | String | True | This is the start date for the scoreboard |
| end\_dt | String | True | This is the end date for the scoreboard |

### DELETE:

The scoreboards DELETE request will delete an existing scoreboard from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the scoreboard entity with the provided object\_key value |

## Rewards

### Endpoint: <URL>/Rewards

### GET:

The rewards GET request will return a list of active rewards. This list can be used to create, edit or delete rewards from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the reward with this specific key |

### POST:

The rewards POST request is used to create new rewards into the active system. To successfully create a new reward the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| fk\_user | Integer | True | This is should be the object\_key of the employee who got rewarded |
| name | String | True | This is the name to identify this reward |
| award\_dt | String | True | This is the date on which the award was awarded |

### PUT:

The rewards PUT request is used to update an existing reward of the active system. To successfully update an existing reward the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing reward |
| fk\_user | Integer | True | This is should be the object\_key of the employee who got rewarded |
| name | String | True | This is the name to identify this reward |
| award\_dt | String | True | This is the date on which the award was awarded |

### DELETE:

The rewards DELETE request will delete an existing reward from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the reward entity with the provided object\_key value |

## Referrals

### Endpoint: <URL>/Referrals

### GET:

The referrals GET request will return a list of active referrals. This list can be used to create, edit or delete referrals from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the referral with this specific key |

### POST:

The referrals POST request is used to create new referrals into the active system. To successfully create a new referral the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| fk\_user | Integer | True | This should be the object\_key of the employee who referred this referral |
| fk\_task | Integer | True | This should be the object\_key of the task which was fulfilled by the employee |
| fk\_candidate | Integer | True | This should be the object\_key of the candidate which was referred |
| fk\_scoreboard | Integer | True | This should be the object\_key of the scoreboard for which this score needs to be linked to |

### PUT:

The referrals PUT request is used to update an existing referral of the active system. To successfully update an existing referral the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing referral |
| fk\_user | Integer | True | This should be the object\_key of the employee who referred this referral |
| fk\_task | Integer | True | This should be the object\_key of the task which was fulfilled by the employee |
| fk\_candidate | Integer | True | This should be the object\_key of the candidate which was referred |
| fk\_scoreboard | Integer | True | This should be the object\_key of the scoreboard for which this score needs to be linked to |

### DELETE:

The referrals DELETE request will delete an existing referral from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the referral entity with the provided object\_key value |

## Candidates

### Endpoint: <URL>/Candidates

### GET:

The candidates GET request will return a list of active candidates. This list can be used to create, edit or delete candidates from the system.

The following data can be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | False | This will only retrieve the candidate with this specific key |

### POST:

The candidates POST request is used to create new candidates into the active system. To successfully create a new candidate the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| referred\_at | String | True | This is the date at which the referral was referred |
| fk\_profile | Integer | False | This is the object\_key property of a profile entity |
| profile | Object <Profile> | False | If the profile field is provided and filled with a valid profile object, the API will handle the creation of this new object and link It to the user being created |

### PUT:

The candidates PUT request is used to update an existing candidate of the active system. To successfully update an existing candidate the mandatory fields must be provided correctly.

The following fields can be added as FormData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| Object\_key | Integer | True | The object\_key value of an existing candidate |
| referred\_at | String | True | This is the date at which the referral was referred |
| fk\_profile | Integer | False | This is the object\_key property of a profile entity |
| profile | Object <Profile> | False | If the profile field is provided and filled with a valid profile object, the API will handle the creation of this new object and link It to the user being created |

### DELETE:

The candidates DELETE request will delete an existing candidate from the active system.

The following data needs to be added as QueryData:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Value type | Mandatory | Information |
| object\_key | Integer | True | This will try to delete the candidate entity with the provided object\_key value |