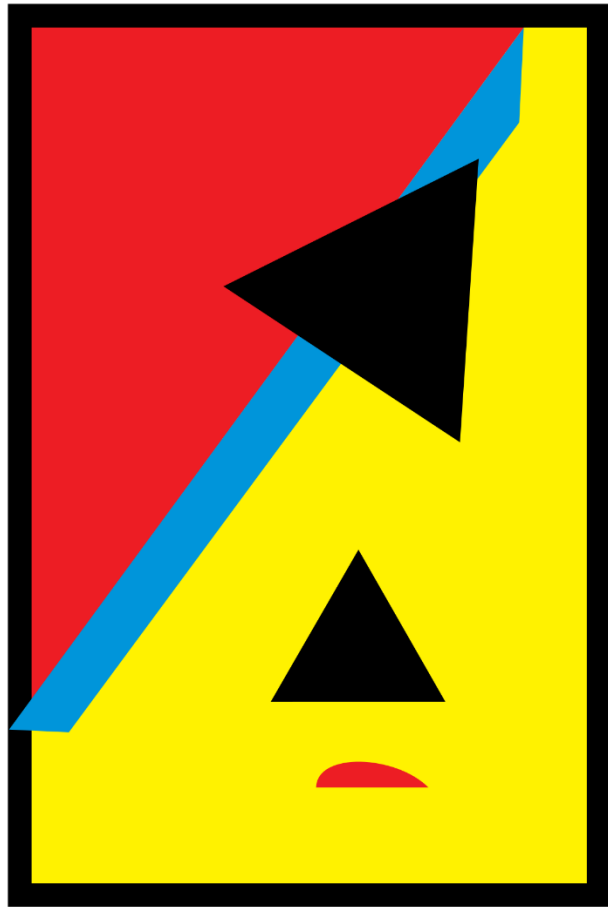


# API Documentation

Referral Application Recruitment



**A L T E N**

## Contents

Document guide .....	4
Version.....	4
Authors .....	4
API Testing .....	5
Endpoints.....	6
Systems .....	6
Endpoint: <URL>/Systems.....	6
/Systems/SeedDatabase .....	6
/Systems/Unit test .....	6
Authentication .....	7
Endpoint: <URL>/Authentication.....	7
POST:.....	7
PUT:.....	7
Users .....	8
Endpoint: <URL>/Users .....	8
GET: .....	8
POST:.....	8
PUT:.....	9
DELETE: .....	9
Profiles .....	10
Endpoint: <URL>/Profiles .....	10
GET: .....	10
POST:.....	10
PUT:.....	11
DELETE: .....	11
Tasks.....	12
Endpoint: <URL>/Tasks .....	12
GET: .....	12
POST:.....	12
PUT:.....	13
DELETE: .....	13
Scoreboards .....	14
Endpoint: <URL>/Scoreboards .....	14

GET:.....	14
POST:.....	14
PUT:.....	15
DELETE:.....	15
Rewards .....	16
Endpoint: <URL>/Rewards.....	16
GET:.....	16
POST:.....	16
PUT:.....	17
DELETE:.....	17
Referrals.....	18
Endpoint: <URL>/Referrals .....	18
GET:.....	18
POST:.....	18
PUT:.....	19
DELETE:.....	19
Candidates.....	20
Endpoint: <URL>/Candidates .....	20
GET:.....	20
POST:.....	20
PUT:.....	21
DELETE:.....	21

## 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

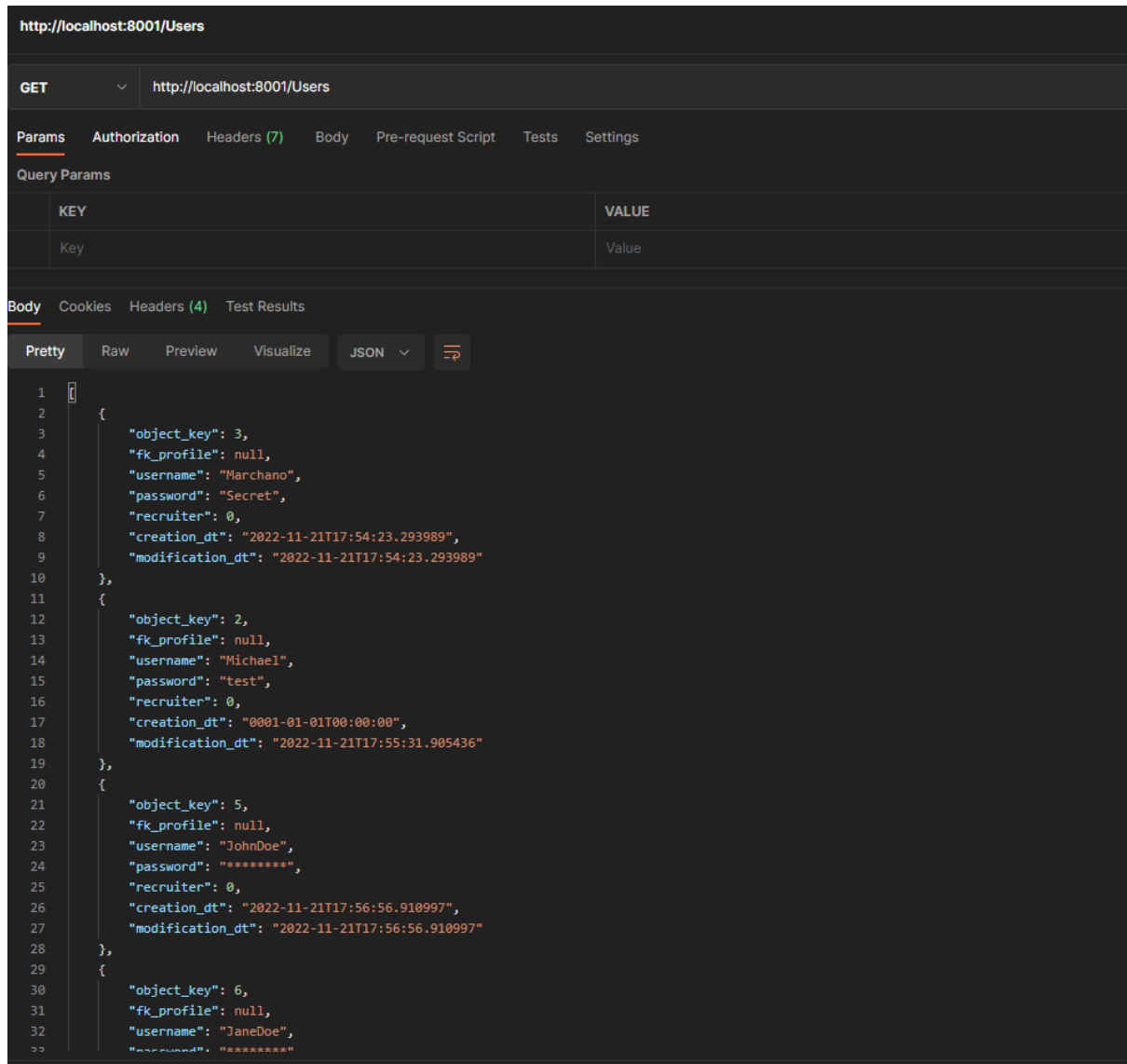
## Authors

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:

```
1  [
2    {
3      "Created": "5",
4      "Get": "Successfull",
5      "Updated": "5",
6      "Delete": "Successfull"
7    },
```

## 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
<b>username</b>	String	True	This is used to find an active user in the system
<b>password</b>	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
<b>fk_user</b>	Integer	True	This is used to find the user
<b>password</b>	String	True	This will be used to set the new password
<b>password_check</b>	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
<b>object_key</b>	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
<b>username</b>	String	True	This will be the login and display name of the user
<b>password</b>	String	True	This password will be used to log into the system if combined with the correct username
<b>recruiter</b>	Integer	True	This field should be either 0 or 1 depending on if this user should be a recruiter or an employee
<b>fk_profile</b>	Integer	False	This is the object_key property of a profile entity
<b>profile</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing user
<b>username</b>	String	True	This will be the login and display name of the user
<b>password</b>	String	True	This password will be used to log into the system if combined with the correct username
<b>recruiter</b>	Integer	True	This field should be either 0 or 1 depending on if this user should be a recruiter or an employee
<b>fk_profile</b>	Integer	False	This is the object_key property of a profile entity
<b>profile</b>	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
<b>object_key</b>	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
<b>object_key</b>	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
<b>name</b>	String	False	This is the first name for the profile entity
<b>surname</b>	String	False	This is the surname for the profile entity
<b>email</b>	String	False	This is the email for the profile entity
<b>phone_number</b>	String	False	This is the phone number for the profile entity
<b>address</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing profile
<b>name</b>	String	False	This is the first name for the profile entity
<b>surname</b>	String	False	This is the surname for the profile entity
<b>email</b>	String	False	This is the email for the profile entity
<b>phone_number</b>	String	False	This is the phone number for the profile entity
<b>address</b>	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
<b>object_key</b>	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
<b>object_key</b>	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
<b>name</b>	String	True	This is the name for the task entity
<b>points</b>	Integer	True	This is the amount of points a task can pay-out upon completion
<b>description</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing task
<b>name</b>	String	False	This is the name for the task entity
<b>points</b>	Integer	False	This is the amount of points a task can pay-out upon completion
<b>description</b>	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
<b>object_key</b>	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
<b>object_key</b>	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
<b>fk_user</b>	Integer	True	This is should be the object_key of the creator of the scoreboard
<b>name</b>	String	True	This is the name for the scoreboard entity
<b>start_dt</b>	String	True	This is the start date for the scoreboard
<b>end_dt</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing scoreboard
<b>fk_user</b>	Integer	True	This is should be the object_key of the creator of the scoreboard
<b>name</b>	String	True	This is the name for the scoreboard entity
<b>start_dt</b>	String	True	This is the start date for the scoreboard
<b>end_dt</b>	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
<b>object_key</b>	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
<b>object_key</b>	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
<b>fk_user</b>	Integer	True	This is should be the object_key of the employee who got rewarded
<b>name</b>	String	True	This is the name to identify this reward
<b>award_dt</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing reward
<b>fk_user</b>	Integer	True	This is should be the object_key of the employee who got rewarded
<b>name</b>	String	True	This is the name to identify this reward
<b>award_dt</b>	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
<b>object_key</b>	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
<b>object_key</b>	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
<b>fk_user</b>	Integer	True	This should be the object_key of the employee who referred this referral
<b>fk_task</b>	Integer	True	This should be the object_key of the task which was fulfilled by the employee
<b>fk_candidate</b>	Integer	True	This should be the object_key of the candidate which was referred
<b>fk_scoreboard</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing referral
<b>fk_user</b>	Integer	True	This should be the object_key of the employee who referred this referral
<b>fk_task</b>	Integer	True	This should be the object_key of the task which was fulfilled by the employee
<b>fk_candidate</b>	Integer	True	This should be the object_key of the candidate which was referred
<b>fk_scoreboard</b>	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
<b>object_key</b>	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
<b>object_key</b>	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
<b>referred_at</b>	String	True	This is the date at which the referral was referred
<b>fk_profile</b>	Integer	False	This is the object_key property of a profile entity
<b>profile</b>	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
<b>Object_key</b>	Integer	True	The object_key value of an existing candidate
<b>referred_at</b>	String	True	This is the date at which the referral was referred
<b>fk_profile</b>	Integer	False	This is the object_key property of a profile entity
<b>profile</b>	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
<b>object_key</b>	Integer	True	This will try to delete the candidate entity with the provided object_key value