Workshop Testen



# Business Case and Requirements

Fontys wants to create a small dating app for students to interact with each other more and find each other. Primarily, this is done based on interest. A prototype REST API has been built and is ready to be delivered, but Fontys insisted that it should be tested first by a group of testers. And you are one of them! Isn’t that straight up fun?

The prototype is very basic and simple and it should do the following when you go to the following urls:

Url: /user/test

Params: none

Body: none

Response: A generic test user. This has been hardcoded to give an insight in the JSON structure.

Exception: none

Url: /user/create

Params: none

Body: User

Response: The newly created user

Exception: none

Url: /user/login

Params: String email, String password

Body: none

Response: A user if login was successful

Exception: WrongCredentialsException

Url: /user/create

Params: none

Body: User

Response: The newly created user

Exception: none

Url: /user/addInterest

Params: String uuid, String interest

Body: none

Response: The user with its new interest added

Exception: none

Url: /user/byId

Params: String uuid

Body: none

Response: A user

Exception: UserNotFoundException

Url: /user/byInterest

Params: String interest

Body: none

Response: A list of users with the specified interest

Exception: UserNotFoundException

Url: /user/byAge

Params: int age

Body: none

Response: A list of users with the specified age

Exception: UserNotFoundException

Url: /user/byAgeRange

Params: int minAge, int maxAge

Body: none

Response: A list of users whose age is between minAge and maxAge

Exception: UserNotFoundException

Url: /user/addMatch

Params: String uuiduser, String uuidmatch

Body: none

Response: The user with the newly added matching user

Exception: UserNotFoundException

Url: /user/matches

Params: String uuid

Body: none

Response: A list of users that are considered a match for the input user

Exception: UserNotFoundException

Url: /users/

Params: none

Body: none

Response: A list of all the existing users

Exception: none

Aside from the regulations for each url, there are a couple more requirements. For instance, each user should have a unique email address and a unique password. Otherwise users might login and accidentally get the wrong account or exceptions from the server. Furthermore, when the server shuts down for whatever reason, the data may not be lost. Lastly, the password has to comply to the following rules in order to be accepted: it has to be at least 8 characters long and must contain upper case, lower case, numbers, and special characters. The user from /user/test is the only exception, because this is a hardcoded used and only serves as a test user to give an insight in the JSON layout of a user.

Be sure to have Postman installed to test the API. Now, it is up to you to see if you can find bugs in the software. ☺