Comments:

1. HTTP Endpoint Documentation - HTTP request methods - HTTP | MDN
2. Refer to helper source files in https://github.com/DanielSeldura/APPSDEV-RESOURCE under user.resource
3. Under no circumstances must a retrieval of user data include the password in the returned data.
4. To make use of the automated UID generation, install the uuid package from https://www.npmjs.com/package/uuid by following the instructions in the package documentation.
   Then install typescript definitions by doing npm i --save-dev @types/uuid in your 3-nest folder

Parameters:

1. **Register a user**
   a. HTTP Method
      POST
   b. Endpoint
      user/register
   c. Body schema
      USER_REGISTER_BODY
   d. Return schema
      USER_RETURN with data of USER DATA
      or
      Error message
   e. Comments
      i. Creates a user and saves it to the "database"
      ii. Fails if the payload
         1. has attributes of the wrong type
         2. is missing an attribute
         3. has an invalid attribute key
         4. Email already exists in database
2. **Retrieve all user data**
   a. HTTP Method

GET

b. Endpoint
   user/all
c. Body schema
   No body
d. Return schema
   USER_RETURN with data of array of USER_DATA || empty array
e. Comments
   i. Retrieves all user data of all users

3. Retrieve a user's data

a. HTTP Method
   GET
b. Endpoint
   user/:id
c. Body schema
   No body
d. Return schema
   USER_RETURN with data of USER_DATA *or* error message
e. Comments
   i. Retrieves a user's data
   ii. Fails if parameter id does not match any users in database

4. Search for a term matching any attribute

a. HTTP Method
   GET
b. Endpoint
   user/search/:term
c. Body schema
   No body
d. Return schema
   array of USER_DATA || empty array
e. Comments
   i. Retrieves a user's data
   ii. Fails if parameter id does not match any users in database

5. Replace all values for the user(with one exception)

a. HTTP Method
   PUT
b. Endpoint
   user/:id
c. Body schema
   USER_REGISTER_BODY
d. Return schema
   USER_RETURN
e. Comments
   i. Does not replace the generated id
   ii. Fails if the payload
      1. has attributes of the wrong type
      2. has an invalid attribute key
      3. is missing an attribute
      4. an email already exists in database that is not of the current user

6. Replace some values in the user(with one exception)

a. HTTP Method
   PATCH
b. Endpoint
   user/:id
c. Body schema
   Any or all of USER_REGISTER_BODY
d. Return schema
   USER_RETURN
e. Comments
   i. Does not replace the generated id
   ii. Fails if the payload
      1. has attributes of the wrong type
      2. has an invalid attribute key
      3. an email already exists in database that is not of the current user

7. Delete a user

a. HTTP Method
   DELETE
b. Endpoint
   user/:id
c. Body schema
   LOGIN_SCHEMA
d. Return schema
   USER_RETURN
e. Comments
   i. Does not replace the generated id
   ii. Fails if the payload
      1. has attributes of the wrong type
      2. has an invalid attribute key
      3. an email already exists in database that is not of the current user