User Microservice API

Overview

This microservice handles the registration, updating, deletion, and retrieving of users.

Paths

GET `/users/{id}/profile`

Get a user profile by ID

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
401	Not authorized to access the user profile *Content* /	No Links
404	User profile does not exist. (May not be a member) *Content* /	No Links
200	Successful retrieval of user profile *Content* /	No Links

PUT `/users/{id}/profile`

Update user profile by user ID

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
200	OK	No Links

PUT `/users/password-reset`

Reset user password

Responses

Code	Description	Links
	default response	No Links
200	Password successfully reset.	No Links

GET `/users/current/profile`

Get a user profile of current logged-in user

Responses

Code	Description	Links
401	Not authorized to access the user profile *Content* /	No Links
404	User profile does not exist. (May not be a member) *Content* /	No Links
200	Successful retrieval of user profile *Content* /	No Links

PUT `/users/current/profile`

Update current logged-in user profile

Responses

Code	Description	Links
200	OK	No Links

PUT `/users/current/profile/status`

Disable/Enabled current logged-in user profile

Responses

Code	Description	Links
200	OK	No Links

GET `/users/current/avatar`

Get avatar image for a user

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/users/current/avatar`

Update avatar image for a user

Responses

Code	Description	Links
200	OK	No Links

POST `/users/registration`

Create a new user registration

Responses

Code	Description	Links
502	The user registration confirmation email was not sent. *Content* /	No Links
201	User has been successfully registered. *Content* /	No Links

Code	Description	Links
400	The user registration DTO contained bad data. *Content* /	No Links
409	There was a data conflict when creating the user. *Content* /	No Links

POST `/users/password-reset-otp`

Create a one-time passcode to reset a user's password

Responses

Code	Description	Links
404	User to create the OTP for was not found.	No Links
422	One-time passcode was not sent through either SMS or Email.	No Links
200	Password OTP is created.	No Links

POST `/users/otp-authentication`

Verify the OTP and allow it to be used.

Responses

Code	Description	Links
200	One-time passcode was successfully verified.	No Links
401	One-time passcode was not correct and was not verified.	No Links

POST `/users/confirmation`

Confirm user registration

Responses

Code	Description	Links
404	User or registration was not found. *Content* /	No Links

Code	Description	Links
200	User registration was successfully confirmed. *Content* /	No Links
410	Token does not exist or is expired. *Content* /	No Links

GET `/users`

Get a paginated response of users

Parameters

Type	Name	Description	Schema
header	**Authorization** *required*		string
query	**search** *optional*		string
query	**pageable** *required*		Pageable

Responses

Code	Description	Links
200	Paginated response was sent. It may have an empty content array which means there are no users. *Content* /	No Links

GET `/users/{id}`

Get a user by ID

Parameters

Type	Name	Description	Schema
path	**id** *required*		integer (int64)

Responses

Code	Description	Links
404	User does not exist. *Content* /	No Links
200	User with specified ID found. *Content* /	No Links

GET `/users/current`

Get the current authenticated user.

Responses

Code	Description	Links
401	Not authorized to access the user. *Content* /	No Links
200	Retrieved current authenticated user. *Content* /	No Links

GET `/health`

Responses

Code	Description	Links
200	OK	No Links

Components

Schemas

UserProfileUpdate

Properties

Name	Description	Schema
username *optional*		string

Name	Description	Schema
email *optional*		string
lastName *optional*		string
phone *optional*		string
driversLicense *optional*		string
income *optional*		integer (int32)
address *optional*		string
city *optional*		string
state *optional*		string
zipcode *optional*		string
mailingAddress *optional*		string
mailingCity *optional*		string
mailingState *optional*		string
mailingZipcode *optional*		string

ResetPasswordRequest

Properties

Name	Description	Schema
otp *optional*	**Maximum Length**: 6 **Minimum Length**: 6	string
username *optional*		string
newPassword *optional*		string

UserAvatarRequest

Properties

Name	Description	Schema
pic *optional*		string

AdminUserRegistration

MemberUserRegistration

UserRegistration

Properties

Name	Description	Schema
username *required*		string
password *required*		string
role *required*		string

UserResponse

Properties

Name	Description	Schema
id *optional*		integer (int64)
firstName *optional*		string
lastName *optional*		string
username *optional*		string
email *optional*		string
role *optional*		enum (MEMBER,EMPLOYEE,ADMINISTRATOR)
enabled *optional*		boolean
memberId *optional*		integer (int64)
membershipId *optional*		string

ResetPasswordAuthentication

Properties

Name	Description	Schema
username *optional*		string
contactMethod *required*		enum (phone,email)

OtpAuthentication

Properties

Name	Description	Schema
username *optional*		string
otp *optional*	**Maximum Length**: 6 **Minimum Length**: 6	string

ConfirmUserRegistration

Properties

Name	Description	Schema
token *optional*		string

ConfirmUserRegistrationResponse

Properties

Name	Description	Schema
username *optional*		string
confirmedAt *optional*		string (date-time)
enabled *optional*		boolean

Pageable

Properties

Name	Description	Schema
offset *optional*		integer (int64)
sort *optional*		Sort
pageNumber *optional*		integer (int32)

Name	Description	Schema
pageSize *optional*		integer (int32)
unpaged *optional*		boolean
paged *optional*		boolean

Sort

Properties

Name	Description	Schema
sorted *optional*		boolean
unsorted *optional*		boolean
empty *optional*		boolean

PageUserResponse

Properties

Name	Description	Schema
totalElements *optional*		integer (int64)
totalPages *optional*		integer (int32)
size *optional*		integer (int32)
content *optional*		< UserResponse > array
number *optional*		integer (int32)
sort *optional*		Sort
pageable *optional*		Pageable
numberOfElements *optional*		integer (int32)
first *optional*		boolean
last *optional*		boolean
empty *optional*		boolean

AddressResponse

Properties

Name	Description	Schema
address *optional*		string
city *optional*		string
state *optional*		string
zipcode *optional*		string

ContactInfo

Properties

Name	Description	Schema
phone *optional*		string
email *optional*		string

UserProfile

Properties

Name	Description	Schema
username *optional*		string
firstName *optional*		string
middleName *optional*		string
lastName *optional*		string
membershipId *optional*		string
income *optional*		integer (int32)
contactInfo *optional*		ContactInfo
billingAddress *optional*		AddressResponse
mailingAddress *optional*		AddressResponse