Introduction

UserAPI

Verwaltet Nutzer (rollenbasiert)

Version:	1.0.0
Contact:	**email**: you@your-company.com
License:	**name**: Apache 2.0
License:	**url**: http://www.apache.org/licenses/LICENSE-2.0.html

Generated on: Sun Jan 31 2021 17:18:00 GMT+0100 (Central European Standard Time)

1. Paths

1.1. put /registerUser

1.2. post /loginToken

1.3. get /user/{userID}

1.4. get /getRoles

1.5. put /editUser

1.6. get /users

1.7. put /setStudyToken/{userID}

1.8. put /addRole

1. Paths

1.1 put /registerUser

Summary	registers user and save
Description	Valid users are saved and API token are generated for them
Operation Id	registerUser
Produces	
Consumes	

Parameters

Name	In	Description	Required	Type	Format	Collection Format	Default	Min	Max
name	query	Username	Yes						
email	query	users mail address	Yes						
password	query	new password for user	Yes						

Tags users admins anyone

Responses

code	description
200	success message / successful registration
400	bad request / bad input / no json request / no user with this email address

1.2 post /loginToken

Summary	create token
Description	if the user inputs the right password, loginToken returns either the existing token or creates a new one
Operation Id	loginToken
Produces	
Consumes	

Parameters

Name	In	Description	Required	Type	Format	Collection Format	Default	Min	Max
email	query	users mail address	Yes						
password	query	input password of the user	Yes						

Tags users admins anyone

Responses

code	description
200	success message

400	bad request / no json request / bad input / no user with this email / wrong password

1.3 get /user/{userID}

Summary	get user
Description	get user by id
Operation Id	user
Produces	
Consumes	

Parameters

Name	In	Description	Required	Type	Format	Collection Format	Default	Min	Max
userID	path	UserId of User to get	Yes						

Tags admins

Responses

code	description
200	success message
400	bad input / no user with this userID

1.4 get /getRoles

Summary	get roles
Description	get all roles
Operation Id	getRoles
Produces	
Consumes	
Parameters	no parameters
Tags	users admins

Responses

code	description
200	success message

1.5 put /editUser

Summary	update user
Description	User will be updated in all properties that got changed
Operation Id	editUser
Produces	
Consumes	

Parameters

Name	In	Description	Required	Type	Format	Collection Format	Default	Min	Max

id	query	id of user to be edited	Yes						
passwordold	query	old password of the user	No						
passwordnew	query	new password of the user	No						
username	query	Name of the user to be edited	Yes						
semester	query	semester of user to be edited	Yes						
study	query	study of user to be edited	No						
role	query	new role of the user, is only changed, if it's the admin who is trying to do it	No						

Tags users admins

Responses

code	description
200	success message
400	bad request / no json request / wrong old password

1.6 get /users

Summary	get users
Description	get a list of all users
Operation Id	users
Produces	
Consumes	
Parameters	no parameters
Tags	admins

Responses

code	description
200	success message

1.7 put /setStudyToken/{userID}

Summary	set study token
Description	Set study token for admin
Operation Id	setStudyToken
Produces	
Consumes	

Parameters

Name	In	Description	Required	Type	Format	Collection Format	Default	Min	Max
userID	path	id of the admin to set the token for	Yes						
study_token	query	token to be set for the admin	Yes						

Tags admins

Responses

code	description
200	success message
400	bad request / no json

1.8 put /addRole

Summary	add role
Description	add a new role to the list of possible roles
Operation Id	addRole
Produces	
Consumes	

Parameters

Name	In	Description	Required	Type	Format	Collection Format	Default	Min	Max
name	query	Name of the role to be added	Yes						

Tags admins

Responses

code	description
200	success message
400	bad request / no json request / bad input / role already exists