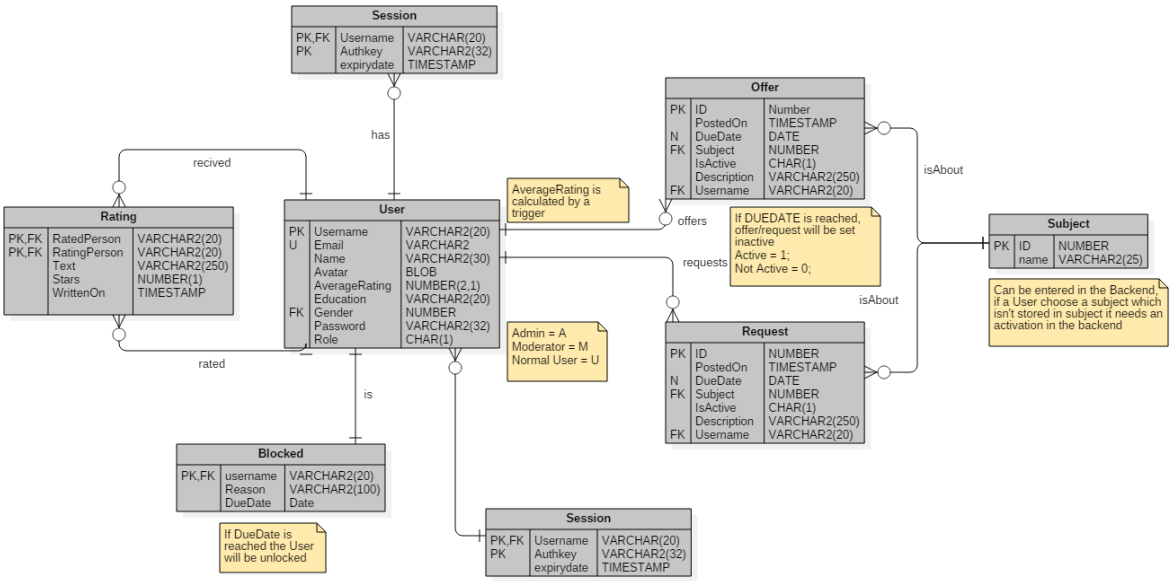
TutoringTrain Webservice V1.1

REST-DOKU

TABLE OF CONTENTS

GENERAL

Data Model



Only JSON format is supported.

All JSON attributes marked in **THIS COLOR** are optional.

Dates format is: "**yyyy-MM-dd'T'hh:mm:ssZ**"
Example: **2017-09-03T09:45:12+0200**

METHODS

[1] SECURITY & AUTHENTICATION

Authentication is managed via session tokens. This is a randomly generated 32-character string which is used to identify the logged in user.

1. Obtaining a session token

Note: A new session token is generated every time this method is called with valid credentials.

A token expires 24 hours after its creation.

Request

Method	URL	Role
POST	url/authentication	None

MediaType	Sample Request Body

application/json	{
	"username": "testuser",
	"password": "098f6bcd4621d373cade4e832627b4f6",
	"requiredRole": "A"
	}

Response

Status	Response
200	654d4e57c9a868dd87759268b225ae44
401	Password and/or username are invalid.
403	User cannot authenticate himself in current context. (eg. normal user tries to log in with admin tool → set requiredRole attribute for this to work)
450	User is currently blocked. (Reason is given in response body)

2. Using the session token

Once a session token has been acquired it can be used to call all other methods. An exception is registering a User as this method does not require any authentication.

For every request the token has to be placed in the “Authorization”-Header in the following format:

“Bearer <token>”

Example:

Key	Value
Authorization	Bearer 4d8068152f82f0ae9ed117365db71cec

A method called without a session token returns **401** HTTP status.

If **401** is returned when a session token was provided it can be assumed that the token has expired or is invalid. In this case you can try to acquire a new token.

If **403** is returned the user associated with the session key has insufficient privileges to perform that method.

If **450** is returned the user associated with the session key has been blocked.

If **451** is returned, the provided JSON-text is invalid. Further details might be given in the returned message.

If **500** is returned an unknown error happened. Further details might be given in the returned message.

These response codes apply to all methods described below but are not mentioned there. Only method-specific response codes are listed.

3. User roles

This System currently supports three roles:

User (U), Moderator (M) and Admin (A)

These roles extend the privileges of their inferior roles, this means a Moderator can perform all actions a User can, and an Admin can perform all actions a Moderator can.

[2] User Resource

1. Register User

Request

Method	URL	Role
POST	url/user/register	None

MediaType	Sample Request Body

application/json	{
	"name": "Kevin Paul",
	"username": "kevin",
	"email": "kevin.paul@gmx.at",
	"password": "098f6bcd4621d373cade4e832627b4f6",
	"education": "HTL",
	"gender": {
	"id": 2
	}
	}

Response

Status	Response

200	{
	"username": "kevin",
	"role": "U",
	"email": "kevin.paul@gmx.at",
	"name": "Kevin Paul",
	"education": "HTL",
	"gender": {
	"id": 2,
	"name": "Male"
	}
	}

409	Username already exists.

2. Get all Users

Request

Method	URL	Role
GET	url/user/all	Admin

Parameters

Type	Name	Description
Query	start	Start index of offers to get
Query	pageSize	Maximum number of offers

Response

Status	Response

200	[
	{
	"username": "kevin",
	"role": "U",
	"email": "kevin.paul@gmx.at",
	"name": "Kevin Paul",
	"education": "HTL",
	"gender": {
	"id": 2,
	"name": "Male"
	}
	},
	{
	"username": "test",
	"role": "U",
	"email": "santnere@edu.htl-villach.at",
	"name": "wacco",
	"education": "HTL",
	"gender": {
	"id": 2,
	"name": "Male"
	},
	"block": {
	"reason": "test"
	}
	},
	{
	"username": "wacco",
	"role": "A",
	"email": "santnere@edu.htl-villach.at",
	"name": "wacco",
	"education": "HTL",
	"gender": {
	"id": 2,
	"name": "Male"
	}
	}
	]

3. Get own User

Request

Method	URL	Role
GET	url/user	User

Response

Status	Response

200	{
	"username": "wacco",
	"role": "A",
	"email": "santnere@edu.htl-villach.at",
	"name": "wacco",
	"education": "HTL",
	"gender": {
	"id": 2,
	"name": "Male"
	}
	}

4. Update any User

Request

Method	URL	Role
PUT	url/user/update	Admin

MediaType	Sample Request Body

application/json	{
	"username": "kevin",
	"name": "Paul Kevin",
	"email": "coolerkev@gmx.at",
	"password": "9d5e3ecdeb4cdb7acfd63075ae046672",
	"education": "HTL",
	"gender": {
	"id": 1
	}
	}

Response

Status	Response
200	
453	User not found

5. Update own User

Request

Method	URL	Role
PUT	url/user/update/own	User

MediaType	Sample Request Body
application/json	{ "name": "Elias Santler", "email": "santnere@edu.htl-villach.at", "password": "098f6bcd4621d373cade4e832627b4f6",

	"education": "HTL",
	"gender": {
	"id": 1
	}
	}

Response

Status	Response
200	

6. Block User

Request

Method	URL	Role
POST	url/user/block	Admin

MediaType	Sample Request Body

application/json	{
	"username": "test",
	"reason": "test",
	"duedate": "2017-09-03T09:45:12+0200"
	}

Response

Status	Response
200	
456	Error blocking user. (Further details may be provided via body)

7. Unblock User

Request

Method	URL	Role
GET	url/user/unblock/{username}	Admin

Parameters

Type	Name	Description
Path	Username	Username of user to unblock

Response

Status	Response
200	

8. *Get all Genders*

Request

Method	URL	Role
GET	url/user/gender	User

Response

Status	Response

200	[
	{
	"id": 3,
	"name": "Female"
	},
	{
	"id": 2,
	"name": "Male"
	},
	{
	"id": 1,
	"name": "Unspecified"
	}
	]

[3] OFFER RESOURCE

1. Create Offer

Request

Method	URL	Role
POST	url/offer	User

MediaType	Sample Request Body

application/json	{
	"description": "This is just a test offer",
	"subject": {
	"id": 3
	}
	}

Response

Status	Response

200	{
	"id": 1,
	"postedon": "2017-10-04T07:18:39+0200",
	"isactive": "1",
	"description": "This is just a test offer",
	"subject": {
	"id": 1,
	"name": "English"
	},
	"user": {
	"username": "wacco"
	}
	}

452	Subject not found

2. Update Offer

Request

Method	URL	Role
PUT	url/offer	User

MediaType	Sample Request Body

application/json	{
	"id": 1,
	"isactive": "1",
	"description": "This is just a test offer",
	"subject": {
	"id": 1
	}
	}

Response

Status	Response
200	
452	Subject not found
454	Offer not found

3. Get newest Offers

Request

Method	URL	Role
GET	url/offer/new	User

Parameters

Type	Name	Description
Query	start	Start index of offers to get
Query	pageSize	Maximum number of offers

Response

Status	Response

200	[
	{
	"id": 2,
	"postedon": "2017-10-04T07:27:58+0200",
	"isactive": "1",
	"description": "Test offer 2",
	"subject": {
	"id": 1,
	"name": "English"
	},
	"user": {
	"username": "wacco"
	}
	},
	{
	"id": 1,
	"postedon": "2017-10-04T07:18:39+0200",
	"isactive": "0",
	"description": "This is just a test offer",
	"subject": {
	"id": 1,
	"name": "English"
	},
	"user": {
	"username": "wacco"
	}
	}
	]

455	Query parameters were not given

4. Get newest Offers of User

Request

Method	URL	Role
GET	url/offer/new/{username}	User

Parameters

Type	Name	Description
Path	Username	Username of user to get newest offers of
Query	start	Start index of offers to get
Query	pageSize	Maximum number of offers

Response

Status	Response

200	[
	{
	"id": 2,
	"postedon": "2017-10-04T07:27:58+0200",
	"isactive": "1",
	"description": "Test offer 2",
	"subject": {
	"id": 1,
	"name": "English"
	},
	"user": {
	"username": "wacco"
	}
	},
	{
	"id": 1,
	"postedon": "2017-10-04T07:18:39+0200",
	"isactive": "0",
	"description": "This is just a test offer",
	"subject": {
	"id": 1,
	"name": "English"
	},
	"user": {
	"username": "wacco"
	}
	}
	]

453	User not found
455	Query parameters were not given

[4] Subject Resource

1. Create Subject

Request

Method	URL	Role
POST	url/subject	Admin

MediaType	Sample Request Body

application/json	{
	"name": "Mathe"
	}

Response

Status	Response

200	{
	"id": 3,
	"name": "Mathe"
	}

409	Subject name already exists (currently not implemented)

2. Update Subject

Request

Method	URL	Role
PUT	url/subject	Admin

MediaType	Sample Request Body

application/json	{
	"id": 2,
	"name": "Deutsch"
	}

Response

Status	Response
200	
452	Subject not found

3. Delete Subject

Request

Method	URL	Role
DELETE	url/subject/{id}	Admin

Parameters

Type	Name	Description
Path	Id	Id of subject to delete

Response

Status	Response
200	
452	Subject not found

4. Get all Subjects

Request

Method	URL	Role
GET	url/subject	User

Response

Status	Response

200	[
	{
	"id": 2,
	"name": "Deutsch"
	},
	{
	"id": 1,
	"name": "English"
	},
	{
	"id": 3,
	"name": "Mathe"
	}
	]