Gleimnet API

API Dokumentation für das Sozial Netzwerk vom Gleimhaus Halberstadt

Default response content-types: application/json
Schemes: http

Summary

Tag: default

Operation	Description
GET /	Starting Point

Tag: Authentication

Operation	Description
POST /login	Login
DELETE /logout	Logout

Tag: Products

Operation	Description
GET /users	Product Types

Tag: User

Operation	Description
GET /users/{id}	User
GET /users/username/{username}	User
GET /users/my	User

Tag: Conversation

Operation	Description
GET /conversations	User
POST /conversations	Conversation
GET /conversations/{id}	Conversation
POST /conversations/{id}	Conversation
GET /conversations/messages/{id}	Message Object

Tag: Timeline

Operation	Description
GET /timeline	Timeline
POST /timeline	Conversation
GET /timeline/{username}	Timeline
POST /timeline/{username}	Post posten
POST /timeline/message/{messageId}	Kommentar posten

Tag: Friends

Operation	Description
GET /friends	User
POST /friends	Freundschaftsanfrage
GET /friends/my	Freundschaften
GET /friends/my/unconfirmed	Freundschaften
GET /friends/username/{username}	User
DELETE /friends/{id}	Freundschaft beenden
POST /friends/{id}	Freundschaftsanfrge bestätigen

Paths

GET /

Starting Point

DESCRIPTION

Starting Point

RESPONSES

Uses default content-types: `application/json`

200 OK

Welcome to the plot device.

StatusMessage

GET /conversations

User

Tags: Conversation

DESCRIPTION

Very short own User Object

RESPONSES

Uses default content-types: `application/json`

200 OK

Das User Objekt

Conversations

POST /conversations

Conversation

Tags: Conversation

DESCRIPTION

Conversation Object

REQUEST BODY

Username

Username

RESPONSES

Uses default content-types: `application/json`

200 OK

Die gesammte Conversation mit der neuen Nachricht

Conversation

GET /conversations/messages/{id}

Message Object

Tags: Conversation

DESCRIPTION

Message Object

REQUEST PARAMETERS

Name	Description	Type	Data type	
id	Message id.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Einzelne Message

Message

GET /conversations/{id}

Conversation

Tags: Conversation

DESCRIPTION

Conversation Object

REQUEST PARAMETERS

Name	Description	Type	Data type	
id	Conversation id.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Einzelne Conversation

Conversation

POST /conversations/{id}

Conversation

Tags: Conversation

DESCRIPTION

Conversation Object

REQUEST BODY

message content

Content

REQUEST PARAMETERS

Name	Description	Type	Data type	
id	Conversation id.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Die gesammte Conversation mit der neuen Nachricht

Conversation

GET /friends

User

Tags: Friends

DESCRIPTION

Very short own User Object

RESPONSES

Uses default content-types: **application/json**

200 OK

Das User Objekt

VeryShortUser

POST /friends

Freundschaftsanfrage

Tags: Friends

DESCRIPTION

Stelle eine neue Freundschaftsanfrage

REQUEST BODY

Username

Username

RESPONSES

Uses default content-types: `application/json`

200 OK

Die gesammte Conversation mit der neuen Nachricht

Friendship

GET /friends/my

Freundschaften

Tags: Friends

DESCRIPTION

Meine bestätigten Freundschaften

RESPONSES

Uses default content-types: `application/json`

200 OK

Das Object mit den Freundschaften

Friendships

GET /friends/my/unconfirmed

Freundschaften

Tags: Friends

DESCRIPTION

Meine bestätigten Freundschaften

RESPONSES

Uses default content-types: **application/json**

200 OK

Das Object mit den Freundschaften

Friendships

GET /friends/username/{username}

User

Tags: Friends

DESCRIPTION

Alle Freunde des Users

REQUEST PARAMETERS

Name	Description	Type	Data type	
username	Username.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Das Object mit den Freundschaften

Friendships

DELETE /friends/{id}

Freundschaft beenden

Tags: Friends

DESCRIPTION

Die Freundschaftsanfrage oder die Freundschaft beenden

REQUEST PARAMETERS

Name	Description	Type	Data type	
id	Friendship id.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Success Nachricht

StatusMessage

POST /friends/{id}

Freundschaftsanfrge bestätigen

Tags: Friends

DESCRIPTION

Die Freundschaftsanfrage bestätigen

REQUEST BODY

Aktivate boolean

Content

REQUEST PARAMETERS

Name	Description	Type	Data type	
id	Friendship id.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Freundschaft

Friendship

POST /login

Login

Tags: Authentication

DESCRIPTION

Der User kann sich hier authentifizien. Gibt ein Objekt mit dem http Basic Auth Token zurück.

REQUEST BODY

Auth

RESPONSES

Uses default content-types: `application/json`

200 OK

Login Object

AuthToken

DELETE /logout

Logout

Tags: Authentication

DESCRIPTION

Logout

RESPONSES

Uses default content-types: `application/json`

200 OK

Logout Nachricht

StatusMessage

GET /timeline

Timeline

Tags: Timeline

DESCRIPTION

Gibt die die eigene Timeline

RESPONSES

Uses default content-types: `application/json`

200 OK

Die eigene Timeline

Timeline

POST /timeline

Conversation

Tags: Timeline

DESCRIPTION

Neuer Post an der Timeline

REQUEST BODY

Inhalt des neuen Posts

Content

RESPONSES

Uses default content-types: `application/json`

200 OK

Die gesammte Conversation mit der neuen Nachricht

Timeline

POST /timeline/message/{messageId} Kommentar posten

Tags: Timeline

DESCRIPTION

Neuer Kommentar an der Timeline

REQUEST BODY

Inhalt des neuen Posts

Content

REQUEST PARAMETERS

Name	Description	Type	Data type	
messageId	Message id	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Der Post mit dem neuen Nachricht

Post

GET /timeline/{username}

Timeline

Tags: Timeline

DESCRIPTION

Gibt die die eigene Timeline

REQUEST PARAMETERS

Name	Description	Type	Data type	
username	Username.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Die eigene Timeline

Timeline

POST /timeline/{username}

Post posten

Tags: Timeline

DESCRIPTION

Neuer Post an der Timeline

REQUEST BODY

Inhalt des neuen Posts

Content

REQUEST PARAMETERS

Name	Description	Type	Data type	
username	Username.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Die gesammte Conversation mit der neuen Nachricht

Timeline

GET /users

Product Types

Tags: Products

DESCRIPTION

The Products endpoint returns information about the *Uber* products offered at a given location. The response includes the display name and other details about each product, and lists the products in the proper display order.

REQUEST PARAMETERS

Name	Description	Type	Data type	
latitude	Latitude component of location.	query	*number* (double)	**required**
longitude	Longitude component of location.	query	*number* (double)	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

An array of products

ITEMS

User

default

Unexpected error

Error

GET /users/my

User

Tags: User

DESCRIPTION

Very short own User Object

RESPONSES

Uses default content-types: **application/json**

200 OK

Das User Objekt

VeryShortUser

GET /users/username/{username}

User

Tags: User

DESCRIPTION

User Object

REQUEST PARAMETERS

Name	Description	Type	Data type	
username	Username.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Das User Objekt

UserWithFriends

GET /users/{id}

User

Tags: User

DESCRIPTION

User Object

REQUEST PARAMETERS

Name	Description	Type	Data type	
id	User id.	path	*string*	**required**

RESPONSES

Uses default content-types: **application/json**

200 OK

Das User Objekt

User

Schema definitions

Auth: *object*

PROPERTIES

username: *string*
d

password: *string*

AuthToken: *object*

PROPERTIES

user: *object*

> **PROPERTIES**
>
> **_id:** *string*
>
> **username:** *string*

session: *object*

> **PROPERTIES**
>
> **userId:** *string*
>
> **key:** *string*
>
> **time:** *string*
>
> **_id:** *string*

authHeader: *string*

Comment: *object*

PROPERTIES

_id: *string*

author: *string*

timeCreated: *string*

content: *string*

Content: *object*

PROPERTIES

content: *string*

Conversation: *object*

PROPERTIES

_id: *string*

timeCreated: *string*

authors: *object[]*

> **ITEMS**
>
> *ObjectId*

messages: *object[]*

> **ITEMS**
>
> *Message*

Conversations: *object*

PROPERTIES

conversations: *object[]*

> **ITEMS**
>
> *Conversation*

numberofItems: *number*

Error: *object*

PROPERTIES

code: *integer* **(int32)**

message: *string*

fields: *string*

Friendship: *object*

PROPERTIES

_id: *string*

isActive: *boolean*
Status der Freundschaftsanfrage

friends: *object[]*

> **ITEMS**
>
> *ObjectId*

Friendships: *object*

PROPERTIES

friendships: *object[]*

> **ITEMS**
>
> *Friendship*

Message: *object*

PROPERTIES

_id: *string*

author: *string*

timeCreated: *string*

content: *string*

ObjectId: *object*

PROPERTIES

_id: *string*
Unique identifier representing a Mongo Dokument

Post: *object*

PROPERTIES

_id: *string*

author: *string*

timeCreated: *string*

content: *string*

comments: *object[]*

> **ITEMS**
>
> *Comment*

shortUser: *object*

PROPERTIES

_id: *string*
Unique identifier representing a specific User

isActive: *boolean*

username: *string*

givenName: *string*

surname: *string*

nickname: *string*

StatusMessage: *object*

PROPERTIES

message: *string*

Timeline: *object*

PROPERTIES

_id: *string*

timeCreated: *string*

messages: *object[]*

> **ITEMS**
>
> *Post*

User: *object*

PROPERTIES

_id: *string*
Unique identifier representing a specific User

isActive: *boolean*

username: *string*

password: *string*

givenName: *string*

surname: *string*

nickname: *string*

birthdate: *string*

description: *string*

avatar: *string*

titlePicture: *string*

timeline: *string*

Username: *object*

PROPERTIES

username: *string*

UserWithFriends: *object*

PROPERTIES

_id: *string*
Unique identifier representing a specific User

username: *string*

givenName: *string*

surname: *string*

nickname: *string*

birthdate: *string*

description: *string*

avatar: *string*

titlePicture: *string*

timeline: *string*

friends: *object[]*

> **ITEMS**
>
> *object*
>
> > **PROPERTIES**
> >
> > **_id:** *string*
> >
> > **username:** *string*

VeryShortUser: *object*

PROPERTIES

_id: *string*
Unique identifier representing a specific User

username: *string*