GatorShare API

Overview

Backend APIs for the GatorShare app

Version information

Version : 1.0

URI scheme

Host : localhost:8080
BasePath : /api/v1

Paths

Gets the current logged-in member

```
GET /current-user
```

Description

This API returns the username of the currently logged-in Member

Responses

HTTP Code	Description	Schema
200	Success	string
401	Unauthorized	string
403	Forbidden	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Logs in an existing member

```
POST /login
```

Description

This API is used to login a member by using the stored credentials in the database

Parameters

Type	Name	Description	Schema
Body	**member** *required*	Member username and password	models.Member

Responses

HTTP Code	Description	Schema
200	Success	string
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Logs out a currently logged in member

```
POST /logout
```

Description

This API clears the tokens in the cookies and database for the logged in Member

Parameters

Type	Name	Description	Schema
Body	**member** *required*	Member username	models.Member

Responses

HTTP Code	Description	Schema
200	Success	string
401	Unauthorized	string
404	Not Found	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Gets a list of members

```
GET /member
```

Description

Gets a slice of members using the limit and offset parameters, sorts based on the column and order (desc or asc) parameters, and filters based off the search_key parameter

Responses

HTTP Code	Description	Schema
200	Success	string

HTTP Code	Description	Schema
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Updates a members information

```
PUT /member
```

Description

This API updates the field values for the logged-in Member

Parameters

Type	Name	Description	Schema
Body	**member** *required*	Updated member info	models.Member

Responses

HTTP Code	Description	Schema
200	Success	string
400	Bad Request	string
401	Unauthorized	string
404	Not Found	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Deletes a member from the system

```
DELETE /member
```

Description

This API removes the Member entity from the database for the logged-in Member

Parameters

Type	Name	Description	Schema
Body	**member** *required*	Member username	models.Member

Responses

HTTP Code	Description	Schema
200	Success	string
401	Unauthorized	string
404	Not Found	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Handles OPTIONS requests

```
OPTIONS /member
```

Description

This API handles OPTIONS requests for CORS preflight

Responses

HTTP Code	Description	Schema
200	Success	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Gets a member’s info by their username

```
GET /member/{username}
```

Description

This API fetches a Member entity by their unique username

Parameters

Type	Name	Description	Schema
Path	**username** *required*	Username	string

Responses

HTTP Code	Description	Schema
200	Success	string
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Retrieves posts by a specific user

```
GET /member/{username}/posts
```

Description

This API fetches all posts created by a specific member

Parameters

Type	Name	Description	Schema
Body	**member** *required*	Member username	models.Member

Responses

HTTP Code	Description	Schema
200	OK	< models.Post > array
500	Internal Server Error	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Creates a new post

```
POST /post
```

Description

This API creates a new post for the logged-in member

Parameters

Type	Name	Description	Schema
Body	**post** *required*	New post	models.Post

Responses

HTTP Code	Description	Schema
200	OK	models.Post
400	Bad Request	string
401	Unauthorized	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Retrieves posts

```
GET /post
```

Description

Gets a slice of posts using the limit and offset parameters, sorts based on the column and order (desc or asc) parameters, and filters based off the search_key parameter

Responses

HTTP Code	Description	Schema
200	OK	< models.Post > array
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Retrieves a specific post by ID

```
GET /post/{postId}
```

Description

This API fetches a post and its comments by the post ID

Parameters

Type	Name	Description	Schema
Path	**postId** *required*	Post ID	string

Responses

HTTP Code	Description	Schema
200	OK	models.Post
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Updates a post

```
PUT /post/{postId}
```

Description

This API updates a post's content if the logged-in member is the author

Parameters

Type	Name	Description	Schema
Path	**postId** *required*	Post ID	string
Body	**post** *required*	Updated post	models.Post

Responses

HTTP Code	Description	Schema
200	OK	models.Post
400	Bad Request	string
401	Unauthorized	string
403	Forbidden	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Deletes a post

```
DELETE /post/{postId}
```

Description

This API deletes a post by its ID if the logged-in member is the author

Parameters

Type	Name	Description	Schema
Path	**postId** *required*	Post ID	string

Responses

HTTP Code	Description	Schema
200	OK	string
400	Bad Request	string
401	Unauthorized	string
403	Forbidden	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Increments the view count of a post

```
PUT /post/{postId}/increment-views
```

Description

This API increments the view count of a specific post by its ID

Parameters

Type	Name	Description	Schema
Path	**postId** *required*	Post ID	string

Responses

HTTP Code	Description	Schema
200	OK	string
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- post

Registers a new member

```
POST /register
```

Description

This API is used to add a Member entity to the database

Parameters

Type	Name	Description	Schema
Body	**member** *required*	New member	models.Member

Responses

HTTP Code	Description	Schema
201	Created	string
400	Bad Request	string

Consumes

- `application/json`

Produces

- `application/json`

Tags

- member

Definitions

models.Comment

Name	Schema
author *optional*	string
comment_id *optional*	string
content *optional*	string
createdAt *optional*	string

Name	Schema
dislikes *optional*	integer
likes *optional*	integer
post_id *optional*	string

models.Member

Name	Description	Schema
bio *optional*		string
createdAt *optional*		string
csrf_token *optional*		string
disliked_posts *optional*		< models.Post > array
email *optional*		string
liked_posts *optional*	Relationships	< models.Post > array
password *optional*		string
session_token *optional*		string
updatedAt *optional*		string
username *optional*		string

models.Post

Name	Description	Schema
author *optional*		string
comments *optional*		< models.Comment > array

Name	Description	Schema
content *optional*		string
createdAt *optional*		string
disliked_by_members *optional*		< models.Member > array
dislikes *optional*		integer
images *optional*		< string > array
liked_by_members *optional*	Relationships	< models.Member > array
likes *optional*		integer
post_id *optional*		string
title *optional*		string
views *optional*		integer