Twister - NoAuth Social Media API

Handling Requests and Responses

Example Code:

```
fetch(`URL`, {
            method: 'GET/POST',
if POST:      headers: {
              'Content-Type': 'application/json',
            },
if POST   body: JSON.stringify(RELEVANT JS OBJECT),
        })
            .then(res => Handle Response)
```

Example Code for file upload:

```
<form id="imageUploadForm" class="mb-0" action="javascript:uploadImage()" enctype="multipart/form-data">
                            <input id="imageUploadFormSelect" type="file" name="image" />
                            <button type="submit">Upload</button>
                        </form>
```

```
const formData = new FormData(document.getElementById( "imageUploadForm"))
fetch("/api/upload/image", { method: "POST", body: formData })
                .then(res => res.json())
                .then(data => {
                    alert(data.message);
                    if (data.status == 201) {

document.getElementById("imageUploadFormSelect").value = null;
                        closeUpload();
                    }
                })
```

Example Code for handling JSON responses:

```
fetch(etc)
.then(res => res.json())
                        .then(data => {
                            if (data.status == 200) {

document.getElementById(`post${post.id}Like`).innerHTML = `${data.liked
? '<i class="fa-solid fa-heart"></i>' : '<i class="fa-regular
fa-heart"></i>'} ${data.likes}`
                            }
                        })
```

GET

/api/post/get/all/:user

Pass in a username and returns all the posts with relevant details for that user (whether they liked the post or not)

Request

Params:

Variable	Type	Description
user	string	Username

Response

JSON:

Array of Object:

Variable	Type	Description
id	number	Post's id
title	string	Post's title
content	string	Post's content
user	string	Post's creator (Username)
creationDate	number	JS Date
lastEdit	number	JS Date
likes	number	Number of likes
liked	boolean	Whether the user liked the post
comments	number	Number of comments

/api/post/comment/get/:post

Pass in post id and returns all the comments for that post

Request

Params:

Variable	Type	Description
post	number	Post ID

Response

JSON:

Object:

Variable	Type	Description
status	number	post exists : 200 post does not exist : 400
comments	array	array of all comments

comments = Array of Object:

Variable	Type	Description
comment	string	Comment’s Content
user	string	Comment’s Creator
creationdate	number	JS Date

/api/pronoun/get

Returns an array of strings containing all pronouns

Response

Array of String

/api/pronoun/get/:id

Pass in pronoun id and returns specified pronoun

Request

Params:

Variable	Type	Description
id	number	Pronoun ID

Response

JSON:

String

/api/image/get/url/:id

Pass in image id and returns specified image's filename. If image does not exist, returns image id 0 (which should be an image not found icon).

Request

Params:

Variable	Type	Description
id	number	Image ID

Response

JSON:

String

/api/image/get/all

Returns an array of strings containing all image URLs

Response

Array of String

/api/image/user/:username

Pass in a username and returns the image set by that user

Request

Params:

Variable	Type	Description
username	string	Username

Response

JSON:

Object:

Variable	Type	Description
index	number	Image’s ID
source	string	Image’s URL

/api/user/get/:username

Pass in a username and returns details of the user

Request

Params:

Variable	Type	Description
username	string	Username

Response

JSON:

Object:

Variable	Type	Description
static	object	Image's ID
dynamic	object	Image's URL
status	number	200 : user found 500 : user not found

static - Object:

Variable	Type	Description
image	number	Image ID
creation	number	JS Date
pronoun	number	Pronoun ID

dynamic - Object:

Variable	Type	Description
posts	number	Number of posts
likes	number	Number of the user's posts liked
liked	number	Number of posts the user liked

POST

/api/post/like

Pass username and post id to toggle if the user has liked the post or not. Returns updated information for that post.

Request

JSON:

Variable	Type	Description
user	string	Username
post	number	Post ID

Response

JSON for ALL STATUSES:

Variable	Type	Description
status	number	200 : success 400 : Invalid Username 404 : Invalid Post

Additional JSON if {status : 200}:

Variable	Type	Description
liked	boolean	whether the user has liked the post
likes	number	number of likes for that post

Additional JSON if {status : 400} and {status : 404}:

Variable	Type	Description
message	string	Error message if status: 400 : Invalid Username 404 : Array OutOfBounds

/api/post/add

Pass a post's information and add the post to data.json. Will always return 200, recommend getting all posts to verify the new post was added. No response likely means a server error or connection error.

Request

JSON:

Variable	Type	Description
title	string	Post's Title
content	string	Post's Content
user	string	Post's Creator

Response

STATUS: 200

/api/post/comment/add

Pass in post ID, username and comment content and will add the comment to the post in data.json. Will always return 200, recommend getting all comments to verify the new comment was added. No response likely means a server error or connection error.

Request

Params:

Variable	Type	Description
post	number	Post ID
user	string	Username
comment	string	Comment’s Content

Response

STATUS: 200

/api/user/image/set

Pass in username and image id and return updated user information. If user is invalid, returns invalid user dataset. If image ID is invalid, no change will occur.

Request

Params:

Variable	Type	Description
username	string	Username
image	number	Image ID

Response

JSON:

Object:

Variable	Type	Description
static	object	Image's ID
dynamic	object	Image's URL
status	number	200 : user found 500 : user not found

static - Object:

Variable	Type	Description
image	number	Image ID
creation	number	JS Date
pronoun	number	Pronoun ID

dynamic - Object:

Variable	Type	Description
posts	number	Number of posts
likes	number	Number of the user's posts liked
liked	number	Number of posts the user liked

/api/upload/image

Send FormData (image file) and return status with message.

Request

JS FormData Object

Response

JSON:

Variable	Type	Description
status	number	201 : image upload successful 400 : error
message	string	201 : Image uploaded successfully 400 : caught JS error