

Deployed Link

[scriptorium-jonathanjschen913-jonathan-chens-projects-82cdad12.vercel.app](https://scriptorium-jonathanjschen913-jonathan-chens-projects-82cdad12.vercel.app)

## User Profile

### Sign up:

This API endpoint handles the creation of new user accounts. It is a POST-only route that accepts user data, performs validation, and saves the user information in a database.

**Endpoint:** POST /api/users/register

### Request Body:

username (String, Required): Unique username for new user.

password (String, Required): User's password, will be hashed.

email (String, Required): User's email address.

phoneNumber (String, Required): User's phone number.

firstName (String, Required): User's first name.

lastName (String, Required): User's last name.

avatar (Number, Required): A number choice from 1-2 of preset avatars.

role (String, Optional): User's role, either USER or ADMIN (default is USER).

### Example:

```
1  {
2    "username": "azkt",
3    "password": "123",
4    "email": "azkt@gmail.com",
5    "phoneNumber": "1234567890",
6    "firstName": "a",
7    "lastName": "z",
8    "avatar": 2
9  }
```

```
1  {
2    "message": "User created",
3    "user": {
4      "uId": 3,
5      "username": "azkt",
6      "role": "USER"
7    }
8  }
```

## Login

This API endpoint handles user authentication, verifying credentials and returning an access token and refresh token upon successful login.

**Endpoint:** POST /api/users/login

### Request Body:

username (String, Required): The username of the user.

password (String, Required): The user's password.

### Example

```
1  {
2    "username": "azkt",
3    "password": "123"
4  }
```

```
{
  "accessToken": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImF6kt@gmail.com",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImF6kt@gmail.com"
}
```

## View specific user profile

This API endpoint displays information for a specific user by their unique ID.

**Endpoint:** GET /api/users/[id]

### Request Parameters:

id (Number, required): Id of user

```
{
  "uId": 1,
  "username": "az",
  "avatar":
"/Users/alex/Downloads/csc309/Scriptorium/group_7328/PPl/public/avatars/2.png",
  "role": "ADMIN",
  "createdAt": "2024-11-03T19:05:04.349Z",
  "blogPosts": [
    {
      "postId": 1,
      "userId": 1,
      "title": "this is an edited post!",
      "description": "testing the edit blog post feature!",
      "body": "the content you see here are edited!",
      "netvote": 0,
      "isHidden": false,
      "isDeleted": false,
      "createdAt": "2024-11-03T19:23:50.776Z"
    }
  ],
  "codeTemplates": [
    {
      "templateId": 1,
      "userId": 1,
      "title": "my first edited code template!",
      "explanation": "Learn how to add 10 + 5 in Python!",
      "codeId": null,
      "language": "python",
      "isDeleted": false,
      "isHidden": true,
      "createdAt": "2024-11-03T20:14:00.314Z"
    }
  ]
}
```

## Edit your own user profile

This API endpoint allows users to update their profile information. It requires authentication and only allows updates to specified fields.

**Endpoint:** PUT /api/users

### Request Parameters:

userId (Number, Required): The unique identifier of the user (**provided as a query parameter**).

### Headers:

Authorization (String, Required): Bearer token for access authorization.

refresh-token (String, Required): Refresh token for access authorization.

### Request Body:

username (String, Optional): The new username for the user.

email (String, Optional): The new email address for the user.

phoneNumber (String, Optional): The new phone number for the user.

firstName (String, Optional): The new first name of the user.

lastName (String, Optional): The new last name of the user.

avatar (Number, Optional): A number choice from 1-2 of preset avatars.

```
1  {
2    |    "username": "AZKT",
3    |    "avatar": 1
4    |  }
```

```
1  {
2    |    "uId": 3,
3    |    "username": "AZKT",
4    |    "avatar": "/Users/alex/Downloads/csc309/Scriptorium/group_7328/PP1/public/avatars/1.png",
5    |    "createdAt": "2024-11-04T02:06:23.584Z",
6    |    "blogPosts": [],
7    |    "codeTemplates": []
8    |  }
```

## Blog Post

### Creation

#### Create Tags

This API endpoint allows the creation of a new tag for blog posts.

**Endpoint:** POST /api/blogPost/createTags

#### Request Body:

tag (String, Required): The name of the tag (there is a preset of tags to choose from)

```
1 {
2   "id": 2,
3   "tag": "miscellaneous"
4 }
```

```
1 {
2   "tag": "miscellaneous"
3 }
```

#### Create Blog Post

This API endpoint allows an authenticated user to create a new blog post.

**Endpoint:** POST /api/blogPost/create

#### Headers:

Authorization (String, Required): Bearer token for access authorization.

refresh-token (String, Required): Refresh token for access authorization.

#### Request Body:

title (String, Required): The title of the blog post.

description (String, Required): A brief description of the blog post.

body (String, Required): The main content of the blog post.

tags (Array of Strings, Optional): List of tags to associate with the blog post.

```
1 {
2   "title": "post1!",
3   "description": "This is my first post!",
4   "body": "testing out the blog post feature!",
5   "tags": ["educational"]
6 }
```

```
1 {
2   "postId": 5,
3   "userId": 1,
4   "title": "post1!",
5   "description": "This is my first post!",
6   "body": "testing out the blog post feature!",
7   "netvote": 0,
8   "isHidden": false,
9   "isDeleted": false,
10  "createdAt": "2024-11-04T02:31:43.031Z"
11 }
```

## Modification

### Edit

This API endpoint allows an authenticated user to edit an existing blog post.

Endpoint: PUT /api/blogPost/edit

#### Headers:

Authorization (String, Required): Bearer token for access authorization.

refresh-token (String, Required): Refresh token for access authorization.

#### Request Parameters:

blogPostId (Number, Required): The unique identifier of the blog post to be edited

(provided as a query parameter).

#### Request Body:

title (String, Optional): The new title of the blog post.

description (String, Optional): The new description of the blog post.

body (String, Optional): The new main content of the blog post.

```
1  {
2    "title": "this is an edited post!",
3    "description": "testing the edit blog post feature!",
4    "body": "the content you see here are edited!"
5  }
```

```
1  {
2    "postId": 3,
3    "userId": 1,
4    "title": "this is an edited post!",
5    "description": "testing the edit blog post feature!",
6    "body": "the content you see here are edited!",
7    "netvote": 0,
8    "isHidden": false,
9    "isDeleted": false,
10   "createdAt": "2024-11-03T19:34:33.972Z"
11 }
```

## Delete

This API endpoint allows an authenticated user to delete an existing blog post by marking it as deleted.

Endpoint: DELETE /api/blogPost/delete

### Headers:

Authorization (String, Required): Bearer token for access authorization.

refresh-token (String, Required): Refresh token for access authorization.

### Request Parameters:

blogPostId (Number, Required): The unique identifier of the blog post to be deleted

**(provided as a query parameter).**

### Response:

```
1  {
2    "postId": 2,
3    "userId": 1,
4    "title": "post1!",
5    "description": "This is my second post!",
6    "body": "testing out the blog post feature!",
7    "netvote": 0,
8    "isHidden": false,
9    "isDeleted": true,
10   "createdAt": "2024-11-03T19:34:26.144Z"
11 }
```

## Vote

This API endpoint allows an authenticated user to vote on a blog post by either upvoting or downvoting.

Endpoint: POST /api/blogpost/vote

### Headers:

Authorization (String, Required): Bearer token for access authorization.

refresh-token (String, Required): Refresh token for access authorization.

### Request Parameters:

blogPostId (Number, Required): The unique identifier of the blog post to vote on

**(provided as a query parameter).**

### Request Body:

voteType (String, Required): The type of vote. It can be either "upvote" or "downvote".

**(Implementation not working)**

## View

### View all blog posts

This API endpoint retrieves a paginated list of blog posts that are not deleted or hidden.

**Endpoint:** GET /api/blogPost

#### Query Parameters:

page (Number, Optional): The page number to retrieve. Defaults to 1.

limit (Number, Optional): The number of blog posts per page. Defaults to 10.

```
{
  "data": [
    {
      "postId": 5,
      "userId": 1,
      "title": "post1!",
      "description": "This is my first post!",
      "body": "testing out the blog post feature!",
      "netvote": 0,
      "isHidden": false,
      "isDeleted": false,
      "createdAt": "2024-11-04T02:31:43.031Z"
    },
    {
      "postId": 4,
      "userId": 1,
      "title": "post1!",
      "description": "This is my first post!",
      "body": "testing out the blog post feature!",
      "netvote": 0,
      "isHidden": false,
      "isDeleted": false,
      "createdAt": "2024-11-03T20:24:17.399Z"
    }
  ],
  "totalPosts": 2,
  "totalPages": 1,
  "currentPage": 1
}
```



## Search by optional criterion

This API endpoint allows users to search for blog posts based on various criteria such as title, description, tags, and associated code templates.

Endpoint: GET /api/blogPost/search

### Request Body:

- searchTitle (String, Optional): A substring to search for in the blog post titles.
- searchDescription (String, Optional): A substring to search for in the blog post descriptions.
- searchTags (Array, Optional): An array of tags to filter the blog posts.
- wantCodeTemplates (Boolean, Optional): A flag to indicate if only posts with code templates should be retrieved.
- searchCodeTemplateTitle (String, Optional): A substring to search for in the titles of associated code templates.
- page (Number, Optional): The page number for pagination. Defaults to 1.
- pageSize (Number, Optional): The number of blog posts to return per page. Defaults to 10.

The screenshot shows a REST client interface with the following details:

- URL:** http://localhost:3000/api/blogPost/search
- Method:** GET
- Body:**

```
{  "searchTitle": "edited"}
```
- Response (JSON):**

```
{  "posts": [    {      "postId": 1,      "userId": 1,      "title": "this is an edited post!",      "description": "testing the edit blog post feature!",      "body": "the content you see here are edited!",      "netvote": 0,      "isHidden": false,      "isDeleted": false,      "createdAt": "2024-11-03T19:23:50.776Z"    },    {      "postId": 3,      "userId": 1,      "title": "this is an edited post!",      "description": "testing the edit blog post feature!",      "body": "the content you see here are edited!",      "netvote": 0,      "isHidden": false,      "isDeleted": false,      "createdAt": "2024-11-03T19:34:33.972Z"    }  ],  "currentPage": 1,  "totalPages": 1,  "totalItems": 2}
```

## Sort blog posts by rating

This API endpoint retrieves blog posts sorted by their rating (net votes) based on the specified order (ascending or descending).

**Endpoint:** GET /api/blogPost/sortRating

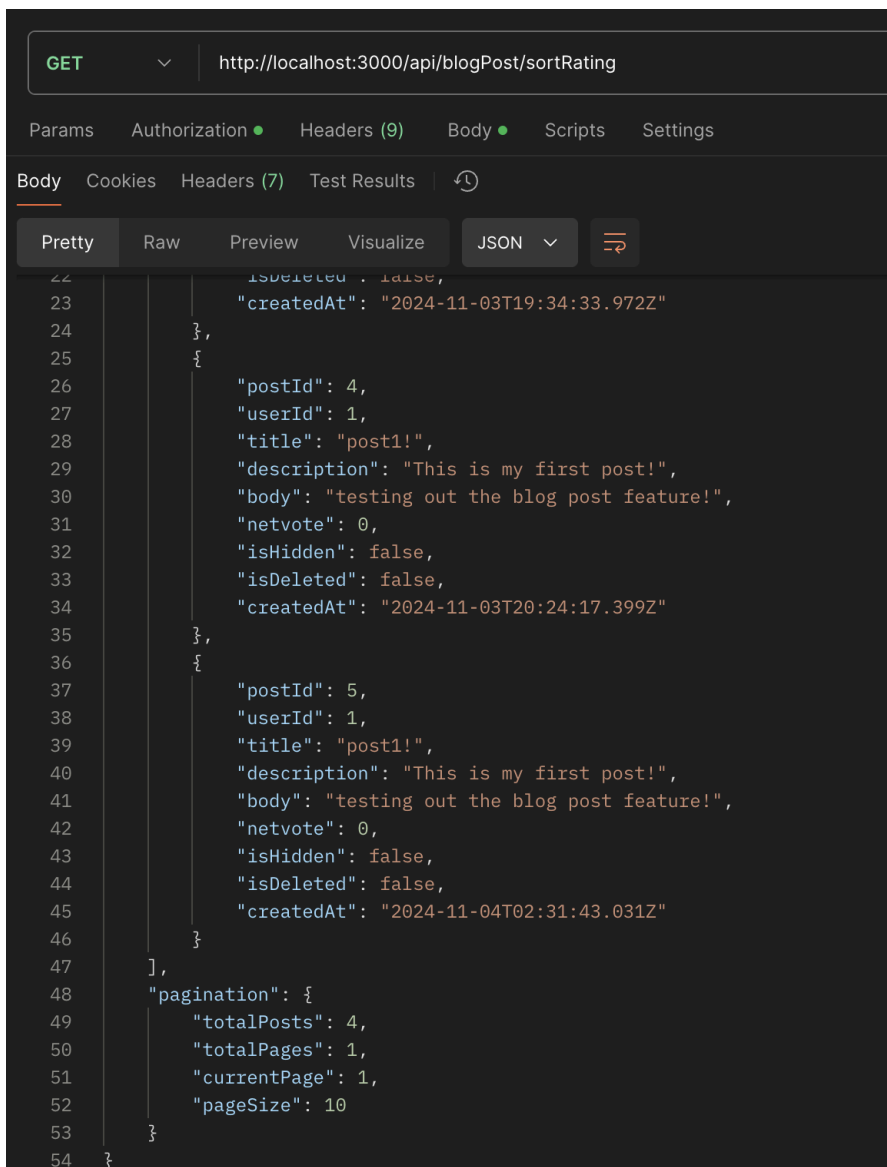
### Request Body:

- sortBy (String, Required): The order in which to sort the blog posts. Acceptable values are asc for ascending and desc for descending.
- blogPostIds (String, Required): A comma-separated list of blog post IDs to be included in the result.

### Query Parameters:

page (Number, Optional): The page number for pagination. Defaults to 1.

pageSize (Number, Optional): The number of blog posts to return per page. Default 10.



```
GET http://localhost:3000/api/blogPost/sortRating

Body
  Cookies
  Headers (7)
  Test Results
  Raw
  Preview
  Visualize
  JSON
  Pretty

[
  {
    "postId": 4,
    "userId": 1,
    "title": "post1!",
    "description": "This is my first post!",
    "body": "testing out the blog post feature!",
    "netvote": 0,
    "isHidden": false,
    "isDeleted": false,
    "createdAt": "2024-11-03T19:34:33.972Z"
  },
  {
    "postId": 5,
    "userId": 1,
    "title": "post1!",
    "description": "This is my first post!",
    "body": "testing out the blog post feature!",
    "netvote": 0,
    "isHidden": false,
    "isDeleted": false,
    "createdAt": "2024-11-04T02:31:43.031Z"
  }
],
{
  "pagination": {
    "totalPosts": 4,
    "totalPages": 1,
    "currentPage": 1,
    "pageSize": 10
  }
}
```

### View specific blog post

This API endpoint retrieves a specific blog post by its ID, checking its visibility and deletion status.

**Endpoint:** GET /api/blogPosts/[id]

#### Query Parameters:

id (Number, Required): The ID of the blog post to be retrieved

The screenshot shows a REST client interface with a GET request to `http://localhost:3000/api/blogPost/4`. The response is displayed in the 'Body' tab, formatted as JSON. The response contains the following data:

```
{
  "postId": 4,
  "userId": 1,
  "title": "post1!",
  "description": "This is my first post!",
  "body": "testing out the blog post feature!",
  "netvote": 0,
  "isHidden": false,
  "isDeleted": false,
  "createdAt": "2024-11-03T20:24:17.399Z"
}
```

## Report

## Code Execution

### Index/default:

This is the “playground” code execution of our app; it is a POST only route that takes in body, language, stdin, and codeId, where body represents the code written by the user but with new lines being represented by \n characters. It returns the stdout of the code if the code did not run into errors; otherwise it returns the stderr.

**Endpoint:** POST /api/codeExec

Example request body:

```
{
  "body": "a = 5\nb = 10\nprint(a + b)",
  "language": "python",
  "stdin": "",
  "codeId": ""
}
```

Example response body:

```
{
  "output": "15\r\n"
}
```

Code objects created using this endpoint will not be associated with any code template objects.

### Create:

This is the endpoint to create a code object that belongs to a code template object. It is a POST only route that takes in body, language, stdin, and codeTemplateId, where body represents the code written by the user but with new lines being represented by \n characters. It turns the json object that represents the code object that was just created.

**Endpoint:** POST /api/codeExec/create

Example request body:

```
{
  "body": "a = 5\nb = 10\nprint(a + b)",
  "language": "python",
  "stdin": "",
  "codeTemplateId": "2"
}
```

Example response body:

```
{
  "code": {
    "codeId": 84,
    "body": "a = 5\nb = 10\nprint(a + b)",
    "language": "python",
    "stdin": "",
  }
}
```

```
        "path":  
        "C:\\Users\\user_here\\a_lot_of_paths_here\\group_7328\\PP1\\generated\\code\\code_173  
0674365198.py",  
        "stdout": null,  
        "stderr": null,  
        "codeTemplateId": 2  
    }  
}
```

## Run:

This is the endpoint to run a code template's associated code object by providing it with the code template's ID. It is a POST-only route that takes in codeTemplateId, and stdin. It returns the stdout of the code if the code did not run into errors; otherwise it returns the stderr.

**Endpoint:** POST /api/codeExec/run

Example request body:

```
{  
    "codeTemplateId": "2"  
}
```

Example response body:

```
{  
    "output": "15\r\n"  
}
```

## Code Template

### Tags creation:

This is the endpoint to create our predefined code template tags that code templates are limited to using. It is a POST-only route that takes in tag as its singular argument. It returns the json object that represents the codeTemplateTag object that just got created.

**Endpoint:** POST /api/codeTemplate/createTags

Example request body:

```
{
  "tag": "testing"
}
```

Example response body:

```
{
  "id": 2,
  "tag": "testing"
}
```

### Creation:

This is the endpoint to create a code template object from scratch. It is a POST-only route that takes in title, explanation, language, and tags as its arguments, where tags are strings that maps to the “tag” attribute of pre-existing codeTemplateTags. It returns the json object that represents the codeTemplate object that just got created

**Endpoint:** POST /api/codeTemplate/create

Example request body:

```
{
  "title": "my first code template",
  "explanation": "adding 10 + 5 in python",
  "language": "python",
  "tags": ["educational"]
}
```

Example response body:

```
{
  "templateId": 4,
  "userId": 1,
  "title": "my first code template",
  "explanation": "adding 10 + 5 in python",
  "codeId": null,
  "language": "python",
}
```

```
    "isDeleted": false,
    "isHidden": false,
    "createdAt": "2024-11-03T23:03:21.796Z"
}
```

## Modification:

This is the endpoint that modifies a pre-existing codeTemplate object. It is a PUT-only route that takes in codeTemplateId from the query and title, explanation, language, and tags from the body as arguments. It returns the json object that represents the edited codeTemplate object.

**Endpoint:** PUT /api/codeTemplate/edit

Example request body:

```
{
  "title": "my first edited code template!",
  "explanation": "Learn how to add 10 + 5 in Python!",
  "tags": ["business"]
}
```

```
{
  "postId": 3,
  "userId": 1,
  "title": "this is an edited post!",
  "description": "testing the edit blog post feature!",
  "body": "the content you see here are edited!",
  "netvote": 0,
  "isHidden": false,
  "isDeleted": false,
  "createdAt": "2024-11-03T19:34:33.972Z"
}
```



View:

GET

http://localhost:3000/api/codeTemplate/1

Send

ParamsAuthorizationHeaders (6)BodyScriptsSettings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (7)Test Results

200 OK292 ms424 B

Save Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "templateId": 1,
3   "userId": 1,
4   "title": "my first code template",
5   "explanation": "adding 10 + 5 in python",
6   "codeId": null,
7   "language": "python",
8   "isDeleted": false,
9   "isHidden": false,
10  "createdAt": "2024-11-03T22:40:51.941Z"
11 }
```

This is a GET-only route at /api/codeTemplate for viewing the json object that represents the code template object with the templateId that was passed through the params. See above for example.

Report

POST

http://localhost:3000/api/codeTemplate/report

Send

ParamsAuthorizationHeaders (8)BodyScriptsSettings

Cookies

noneform-datax-www-form-urlencodedorawbinaryGraphQLJSON

Beautify

1 {
2 "codeTemplateId": 1,
3 "body": "inappropriate",
4 "contentType": "code template"
5 }

BodyCookiesHeaders (7)Test Results

201 Created46 ms412 B

Save Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "reportId": 10,
3   "userId": 1,
4   "body": "inappropriate",
5   "isResolved": false,
6   "contentType": "code template",
7   "createdAt": "2024-11-04T02:48:01.506Z",
8   "blogPostId": null,
9   "commentId": null,
10  "codeTemplateId": 1
11 }
```

This is a POST-only route at `/api/codeTemplate/report` that generates a report object for a code template. It returns the json object that represents the report object. See above for example usage.

## Comment Creation

The screenshot shows a REST client interface with a POST request to `http://localhost:3000/api/comment/create`. The request body is a JSON object: `{ "body": "that is so interesting!", "blogPostId": 1 }`. The response is a 201 Created status with a response time of 117 ms and a body size of 396 B. The response body is a JSON object: `{ "commentId": 1, "userId": 1, "body": "that is so interesting!", "parentId": null, "blogPostId": 1, "depth": 0, "isHidden": false, "isDeleted": false, "createdAt": "2024-11-04T02:52:04.760Z" }`.

```
POST http://localhost:3000/api/comment/create

{
  "body": "that is so interesting!",
  "blogPostId": 1
}
```

201 Created • 117 ms • 396 B

```
{
  "commentId": 1,
  "userId": 1,
  "body": "that is so interesting!",
  "parentId": null,
  "blogPostId": 1,
  "depth": 0,
  "isHidden": false,
  "isDeleted": false,
  "createdAt": "2024-11-04T02:52:04.760Z"
}
```

This is a POST-only route at `/api/comment/create` that creates a top-level comment under a post. It returns a json object that represents the top-level comment object that was just created. See above for usage.

## Modification

The screenshot shows a REST client interface with a PUT request to `http://localhost:3000/api/comment/edit`. The request body is a JSON object: `{ "body": "I am edited", "commentId": 1, "blogPostId": 1 }`. The response is a 200 OK status with a response time of 31 ms and a body size of 378 B. The response body is a JSON object: `{ "commentId": 1, "userId": 1, "body": "I am edited", "parentId": null, "blogPostId": 1, "depth": 0, "isHidden": false, "isDeleted": false, "createdAt": "2024-11-04T02:52:04.760Z" }`.

```
PUT http://localhost:3000/api/comment/edit

{
  "body": "I am edited",
  "commentId": 1,
  "blogPostId": 1
}
```

200 OK • 31 ms • 378 B

```
{
  "commentId": 1,
  "userId": 1,
  "body": "I am edited",
  "parentId": null,
  "blogPostId": 1,
  "depth": 0,
  "isHidden": false,
  "isDeleted": false,
  "createdAt": "2024-11-04T02:52:04.760Z"
}
```

This a PUT-only route at `/api/comment/edit` that edits a comment under a post. It returns a JSON object that represents the top comment object that was just edited. See above for usage.