

Scriptorium

POST Login

`http://localhost:3000/api/auth/login`

This endpoint logs in with an account, the response would be a valid JWT token.

Body raw (json)

json

```
{
  "email": "test5@scriptorium.com",
  "password": "password123"
}
```

POST Login (admin)

`http://localhost:3000/api/auth/login`

Same as above, this is just the default admin account created with `startup.sh`.

Body raw (json)

json

```
{
  "email": "admin@scriptorium.com",
  "password": "admin"
}
```

POST Signup

```
http://localhost:3000/api/auth/signup
```

This endpoint creates an user account with email, password, first name, last name, phone, and an user uploaded avatar. Only USER accounts can be created in this way, admin accounts are created during `startup.sh` through the `createAdmin.js` script.

Note: this endpoint uses form-data to allow for uploading avatars

Body formdata

email	test5@scriptorium.com
password	password123
firstName	Test
lastName	User
phone	123 456 789
avatar	

POST Logout

```
http://localhost:3000/api/auth/logout
```

Logs out of an account by invalidating the JWT token.

HEADERS

Authorization	Bearer
---------------	--------

GET Test Authentication

```
http://localhost:3000/api/auth/testAuth
```

This endpoint is for testing if an account is authenticated (logged in) by checking the JWT token. This endpoint is for ease of testing of, it simply calls the `isAuthenticated` middleware located in `logout.js`. Any endpoint that requires authentication would call `isAuthenticated` too.

HEADERS

Authorization

Bearer

GET Get user by id

```
http://localhost:3000/api/users/1
```

Responds with information of a user specified by user id in query.

PUT Update user by id

```
http://localhost:3000/api/users/2
```

Updates user information, must be authenticated.

HEADERS

Authorization

Bearer

Body `formdata`

email test5@scriptorium.com

firstName Test (EDITED)

lastName User

phone 123 456 789

avatar

DELETE Delete user by id

```
http://localhost:3000/api/users/2
```

Deletes the specified user, requires authentication.

HEADERS

Authorization

Bearer

POST Create blog post

http://localhost:3000/api/posts/

This endpoint creates a blog post. Code templates can be linked to blog post by putting the code template ids in `codeTemplateIds`.

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "title": "My First Blog Post COPY",
  "description": "An introduction to my blog post.",
  "content": "This is the main content of the blog post.",
  "tags": ["Introduction", "FirstPost"],
  "codeTemplateIds": []
}
```

GET Get all my blog posts

http://localhost:3000/api/posts/

Gets all the blog posts authored by the current logged in user.

HEADERS

Authorization

Bearer

GET Get blog post by id

```
http://localhost:3000/api/posts/1
```

Gets a blog post by its id.

HEADERS

Authorization

Bearer

PUT Edit blog post

```
http://localhost:3000/api/posts/1
```

This endpoint edits an existing blog post, requires authentication.

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "title": "My First Blog Post",
  "description": "An introduction to my blog post. (EDIT2)",
  "content": "This is the main content of the blog post.",
  "tags": ["Introduction", "FirstPost"],
  "codeTemplateIds": [1]
}
```

DELETE Delete blog post

```
http://localhost:3000/api/posts/2
```

Deletes existing blog post, requires authentication.

HEADERS

Authorization

Bearer

POST Save code template

`http://localhost:3000/api/users/saveCodeTemp`

Saves a code template, programming language must be specified under `languageId` . (id 1 = JavaScript, id 2 = Python, id 3 = Java, id 4 = C, id 5 = C++) Programming languages are added to the database during `startup.sh` by running the script `addLanguages.js`

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "title": "Template Test",
  "description": "Test template.",
  "code": "print('testing')",
  "languageId": 2, // Assuming 1 is a valid language ID in your database
  "tags": ["Python", "Example"]
}
```

GET Get any code template

`http://localhost:3000/api/users/getAnyTemp?query=Python`

Gets code templates that includes the query input.

PARAMS

query

Python

GET Get my code template

```
http://localhost:3000/api/users/getMyTemp
```

Gets code templates saved by currently logged in user.

HEADERS

Authorization	Bearer
---------------	--------

GET Get template by id

```
http://localhost:3000/api/users/getTemp?id=1
```

Gets the code template by its id.

HEADERS

Authorization	Bearer
---------------	--------

PARAMS

id	1
----	---

POST Fork template

```
http://localhost:3000/api/users/forkTemp
```

Forks an existing template and saves it as a new template. Must be logged in to save forked template.

HEADERS

Authorization	Bearer
---------------	--------

Body raw (json)

json

```
{  
  "templateId": 5  
}
```

DELETE Delete template

http://localhost:3000/api/users/editTemp?id=1

Deletes existing code template, requires authentication.

HEADERS

Authorization	Bearer
---------------	--------

PARAMS

id	1
----	---

PUT Edit template

http://localhost:3000/api/users/editTemp?id=1

Edit code template, requires authentication.

HEADERS

Authorization	Bearer
---------------	--------

PARAMS

id	1
----	---

Body raw (json)

json

```
{
  "title": "Updated Template Title",
  "description": "Updated Template Description",
  "code": "print('testing edit');",
  "tags": ["newTag1", "newTag2"]
}
```

POST Add new comment

http://localhost:3000/api/posts/1/comments

Adds a comment to a blog post. Blog post id is put in the url: `posts/[id]/comments`

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "content": "This is a comment on the blog post."
}
```

POST Reply to comment

http://localhost:3000/api/posts/5/comments

Add a reply to a comment, same as adding a comment to a post but you put `parentCommentId` in the request body.

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "content": "This is a reply to an existing comment.",
  "parentCommentId": 3
}
```

GET Fetch comments on post

http://localhost:3000/api/posts/3/comments

This endpoint responds with all the comments and replies on a given post.

HEADERS

Authorization

Bearer

POST Add or update vote

http://localhost:3000/api/posts/1/votes

This endpoint puts a upvote or downvote on a post or comment, put `commentId` in request body if upvoting a comment and leave it blank otherwise. Putting +1 is an upvote and -1 is a downvote.

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "value": 1,  // +1 for upvote, -1 for downvote
  "commentId": 1
}
```

GET Fetch total votes

http://localhost:3000/api/posts/3/votes

Gets the total number of votes of a post. Total votes is sum of upvote/downvotes, if there is 1 upvote and 1 downvote, the total votes would be 0. Put post id in url.

HEADERS

Authorization Bearer

POST Report content

http://localhost:3000/api/reports

Report content, put contend id and content type in response body. A reason can also be added.

HEADERS

Authorization Bearer

Body raw (json)

json

```
{
  "contentId": 1,  // The ID of the post or comment you're reporting
  "contentType": "post",  // "post" for a blog post or "comment" for a comment
  "reason": "This content is inappropriate and violates guidelines"
}
```

PUT Hide content

http://localhost:3000/api/reports/hide-content

Admin only endpoint. Allows admin to hide content.

HEADERS

Authorization

Bearer

Body raw (json)

json

```
{
  "contentId": 5,
  "contentType": "post"
}
```

GET Fetch reported content

http://localhost:3000/api/reports/reported-content

Admin only endpoint. Shows the admin a list of reported content and report details.

HEADERS

Authorization

Bearer

GET Get post by link (visitor)

http://localhost:3000/api/visitors/fetchFromLink?id=5

Lets a visitor see a post by post id in query.

PARAMS

id

5

GET Get code template (visitor)

```
http://localhost:3000/api/visitors/viewFromLink?id=3
```

Lets visitor see a code template by template id in query.

PARAMS

id

3

GET Mentioned code template

```
http://localhost:3000/api/visitors/mentionedCodeTemplates?id=1
```

Lists all blog posts that links to a specified code template. Put template id in query.

PARAMS

id

1

GET Search blog post

```
http://localhost:3000/api/visitors/searchBlogPost?query=First
```

Searches for all blog posts that contain a specified word in query.

PARAMS

query

First

GET Sorted blog posts

```
http://localhost:3000/api/visitors/blogPostsSorted
```

http://localhost:3000/api/visitors/blogPostsSorted

Lists all blog posts sorted by votes.

GET Sorted comments

http://localhost:3000/api/visitors/commentsSorted?id=1

Lists all comments on a specified blog post sorted by votes. Put post id in query.

PARAMS

id	1
----	---

POST Execute code

http://localhost:3000/api/execute

Allows code execution, must specify language. Input can be provided if needed.

Body raw (json)

json

```
{
  "language": "Java",
  "code": "public class Main { public static void main(String[] args) { System.out.println(
  \"input\": \""
```

POST Execute code template

http://localhost:3000/api/executeTemp

Executes code saved in a code template.

Body raw (json)

json

```
{  
  "templateId": 3, // Replace with a valid template ID in your database  
  "input": ""  
}
```