

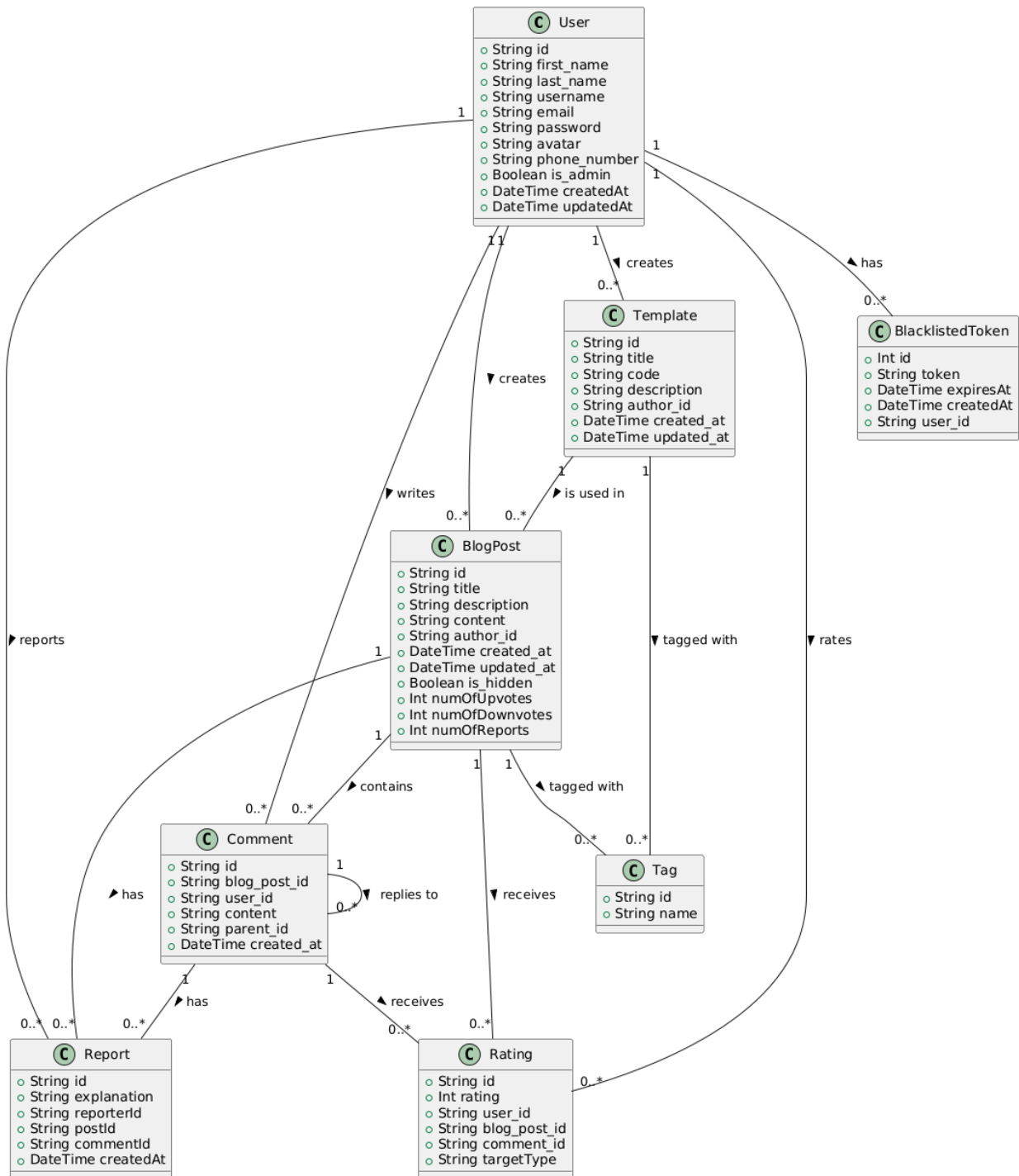
Docs

Credentials

- Admin
 - Username: admin_user
 - Password: admin_password
 - Authentication Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJiNTQ4MmQzZC01MDQ5LTRiMjUtYWFIYy0wMTBiOTU5ZTAxMWYiLCJpYXQiOiJlMzA2NjY5NDYsImV4cCI6NDg4NjQyNjk0Nn0.Wwbk25dCLcwvjCD5AgsNF7hGs3BdWmEu1zzOy51nSw

Model Design

Class Diagram



- User Model

- Each user contains basic details, in addition to a flag `is_admin` which is false by default
- Each user can have multiple template and blog post entries, comment on blog posts, report blog posts, and rate comments or blog posts

- Template Model
 - Each template is associated with an author (User)
 - Have fields like title, code and description
 - Many-to-many relationship with tags
- BlogPost Model
 - Each blog post is associated with an author (User)
 - Has fields like title, description, content
 - Relationship with comments, ratings, reports and tags
 - Has number of upvotes, downvotes and reports
 - Many-to-many relationship with tags
- Comment Model
 - Keeps track of blog_post_id and user_id to associate commenter with post
 - A comment can be added to a comment (replies) with foreign key parent_id
 - Has number of upvotes, downvotes and reports
- Rating Model
 - Manages ratings from blog posts and comments (targetType distinguished them)
- Tag Model
 - Tags are unique by name and are able to be connected to multiple templates or blog posts (many-to-many relationship)
- Report Model
 - Allows users to report issues with blog posts or comments
- Blacklisted Token Model
 - Keeps track of the tokens that have been “blacklisted” to ensure user can’t do authorized actions once they’ve logged out
 - Every entry has token (unique string), expiry date and timestamp of when created
 - Manages user sessions and prevents unauthorized users

API Endpoints

- Endpoint: /api/auth/signup
 - Description: Registers a new user, performing validation checks on the inputted data, creates a new user in the database
 - Allowed Methods: POST
 - Payloads:

First Name: String
 Last Name: String
 Username: String
 Email: String
 Password: String

Phone Number: String
Avatar: String

- Example Request:

Content-Type: application/json

```
{
  "first_name": "John",
  "last_name": "Doe",
  "username": "johndoe",
  "email": "john.doe@example.com",
  "password": "securePassword123",
  "phone_number": "1234567890",
  "avatar": "/uploads/avatars/avatar1.png"
}
```

- Example Response:

```
{
  "message": "User created successfully",
  "user": {
    "first_name": "John",
    "last_name": "Doe",
    "username": "johndoe",
    "email": "john.doe@example.com",
    "avatar": "/uploads/avatars/avatar1.png",
    "phone_number": "1234567890"
  }
}
```

- Endpoint: /api/auth/login

- Description: This endpoint allows users to login using the credentials they created when signing up. There is checking for whether the user uses their email or username to sign in, validation for the password, and the generation of a JWT token for authenticated actions.
- Note: For testing, we have saved the token to ***authToken*** to use for the rest of the Tests
- Allowed Methods: POST
- Payloads:

Identifier: String
Password: String

- Example Request:

Content-Type: application/json

```
{
  "identifier": "john.doe@example.com",
  "password": "securePassword123"
}
```

- Example Response:

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI3Y2Y2YzhmMy1iMTU5LTQ4MWQtYTFjNi0yYTI4ZjExMzZmZmQiLCJpYXQiOiE3MzA2MTcyNzgsImV4cCI6MTczMDYyMDg3OH0.EEQ37hjvBSCctIELIoi5f6MEmhjcYFxr6Y69hNdEOz4"
}
```

- Endpoint: /api/auth/logout

- Description: Endpoint allows users to logout, invalidating the JWT token tied to their current login. The token is added to a makeshift blacklist to ensure it cannot be used in the future.
- Allowed Methods: POST
- Payloads:

// No request body required

- Example Request:

Authorization: Bearer {{authToken}}

- Example Response:

```
{
  "message": "Logout successful"
}
```

- Endpoint: /api/updateUser

- Description: Allows user to update profile information, including username, phone number and avatar
- Allowed Methods: PUT
- Payloads:

Username: String
Phone Number: String
Password: String
Avatar: String

- Example Request:

```
Authorization: Bearer {{authToken}}
Content-Type: application/json

{
  "username": "newusername",
  "phone_number": "1234567890",
  "password": "newsecurepassword",
  "avatar": "/uploads/avatars/avatar2.png"
}
```

- Example Response:

```
{
  "message": "User updated successfully",
  "user": {
    "username": "newusername",
    "phone_number": "1234567890",
    "avatar": "/uploads/avatars/avatar2.png"
  }
}
```

- Endpoint: /api/auth/blog/[id]

- Description: As an authenticated user I want to edit or delete posts by their specific post [id]
- Allowed Methods: PUT, DELETE
- Payloads:

```
Title: String
Description: String
Tags: Tag[]
templateIds: Template[]
```

- Example Request:

```
Authorization: Bearer {{authToken}}
Content-Type: application/json

{
  "title": "Updated Blog Post Title",
  "description": "Updated description of the blog post.",
  "tags": ["UpdatedTag1", "UpdatedTag2"]
}
```

- Example Response:

```
{
  "id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "title": "Updated Blog Post Title",
  "description": "Updated description of the blog post.",
  "content": "Sample content for the blog post.",
  "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
  "created_at": "2024-11-03T16:21:51.931Z",
  "updated_at": "2024-11-03T16:22:29.480Z",
  "numOfUpvotes": 0,
  "numOfDownvotes": 0,
  "numOfReports": 0,
  "tags": [
    {
      "id": "15be47dd-0151-4120-9f0f-b2f3cb8ce567",
      "name": "Tag2"
    },
    {
      "id": "26618c94-78d0-49d0-9875-797bd3540a2a",
      "name": "Tag1"
    },
    {
      "id": "b0e0c1e3-5683-4555-bec5-fcdc094f925b",
      "name": "UpdatedTag1"
    },
    {
      "id": "b3b78415-1251-4611-b893-9fb70ec156f4",
      "name": "UpdatedTag2"
    }
  ],
  "templates": []
}
```

- Endpoint: /api/auth/blog/idx
 - Description: As a user I want to create a blog post or search for all blog Posts posted
 - Allowed Methods: POST, GET
 - Payloads:

```
Title: String
Description: String
Tags: tags[]
Content: String
TemplateIds: template[]
```

- Example Request:

Authorization: Bearer {{authToken}}
Content-Type: application/json

```
{
  "title": "Sample Blog Pwost23",
  "description": "This is a test blog post.",
  "tags": ["Tag1", "Tag2"],
  "content": "Sample content for the blog post."
}
```

- Example Response:

```
{
  "id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "title": "Sample Blog Pwost23",
  "description": "This is a test blog post.",
  "content": "Sample content for the blog post.",
  "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
  "created_at": "2024-11-03T16:21:51.931Z",
  "updated_at": "2024-11-03T16:21:51.931Z",
  "numOfUpvotes": 0,
  "numOfDownvotes": 0,
  "numOfReports": 0,
  "tags": [
    {
      "id": "15be47dd-0151-4120-9f0f-b2f3cb8ce567",
      "name": "Tag2"
    },
    {
      "id": "26618c94-78d0-49d0-9875-797bd3540a2a",
      "name": "Tag1"
    }
  ],
  "templates": [],
  "username": "tester"
}
```

- Endpoint: /api/auth/templates/[id]
 - Description: As a user I want to edit or delete templates by their specific post [id]
 - Allowed Methods: PUT, DELETE
 - Payloads:

Title: String
Description: String
Code: String
Tags: tags[]

- Example Request:

Authorization: Bearer {{authToken}}
Content-Type: application/json

```
{  
  "title": "New Title",  
  "description": "Updated description",  
  "code": "Updated code goes here",  
  "tags": ["tag1", "tag2"]  
}
```

- Example Response:

```
{  
  "id": "7bcc5ae2-c89f-4563-9090-ce7e7bc65a0d",  
  "title": "New Title",  
  "code": "Updated code goes here",  
  "description": "Updated description",  
  "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",  
  "created_at": "2024-11-03T05:05:24.489Z",  
  "updated_at": "2024-11-03T16:43:31.136Z"  
}
```

- Endpoint: /api/auth/blog/[id]
 - Description: As a user I want to edit or delete posts by their specific post [id]
 - Allowed Methods: PUT, DELETE
 - Payloads:

Title: String
Description: String
Tags: tags[]

- Example Request:

Authorization: Bearer {{authToken}}

Content-Type: application/json

```
{
  "title": "Updated Blog Post Title",
  "description": "Updated description of the blog post.",
  "tags": ["UpdatedTag1", "UpdatedTag2"] // New set of template IDs to
  associate
}
```

○ Example Response:

```
{
  "id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "title": "Updated Blog Post Title",
  "description": "Updated description of the blog post.",
  "content": "Sample content for the blog post.",
  "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
  "created_at": "2024-11-03T16:21:51.931Z",
  "updated_at": "2024-11-03T16:22:29.480Z",
  "numOfUpvotes": 0,
  "numOfDownvotes": 0,
  "numOfReports": 0,
  "tags": [
    {
      "id": "b0e0c1e3-5683-4555-bec5-fcdc094f925b",
      "name": "UpdatedTag1"
    },
    {
      "id": "b3b78415-1251-4611-b893-9fb70ec156f4",
      "name": "UpdatedTag2"
    }
  ],
  "templates": []
}
```

- Endpoint: /api/auth/templates/idx
 - Description: As a user I want to create a templates or fetch all templates
 - Allowed Methods: POST, GET
 - Payloads (for POST):

Title: String
Code: String
Description: String
Tags: tags[]

○ Example Request:

Authorization: Bearer {{authToken}}
Content-Type: application/json

```
{
  "title": "bwwrosj",
  "code": "console.log('Hello World');",
  "description": "This template prints Hello World to the console.",
  "tags": ["tag1", "tag2", "Template"]
}
```

○ Example Response:

```
{
  "id": "77b2e88d-9b27-4622-b3f6-3fe121f2a73e",
  "title": "bwwrosj",
  "code": "console.log('Hello World');",
  "description": "This template prints Hello World to the console.",
  "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
  "created_at": "2024-11-03T16:43:25.544Z",
  "updated_at": "2024-11-03T16:43:25.544Z"
}
```

● Endpoint: /api/templates/idx

- Description: As a visitor I want to browse through all templates posted with filters ie. name, tags, code, with pagination
- Allowed Methods: GET
- Payloads:

○ Example Request:

http://localhost:3000/api/templates/idx?search=New Title

○ Example Response:

```

{
  "templates": [
    {
      "id": "7bcc5ae2-c89f-4563-9090-ce7e7bc65a0d",
      "title": "New Title",
      "code": "Updated code goes here",
      "description": "Updated description",
      "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
      "created_at": "2024-11-03T05:05:24.489Z",
      "updated_at": "2024-11-03T16:43:31.136Z",
      "author": {
        "username": "tester"
      },
      "tags": [
        {
          "id": "3056f117-ce64-42a4-b4ad-5903c584860d",
          "name": "tag2"
        },
        {
          "id": "73fc4b98-4299-454a-aa24-ddede1d76c6a",
          "name": "tag1"
        }
      ]
    }
  ],
  "pagination": {
    "totalTemplates": 1,
    "totalPages": 1,
    "currentPage": 1,
    "limit": 10
  }
}

```

- Endpoint: /api/templates/[id]
 - Description: As a visitor I want to modify templates posted
 - Allowed Methods: PUT
 - Payloads:

Title: String
description: string
code: string
tags: tags[]

- Example Request:

```
{  
  "title": "New Title",  
  "description": "Updated description",  
  "code": "Updated code goes here",  
  "tags": ["tag1", "tag2"]  
}
```

- Example Response:

```
{  
  "id": "7bcc5ae2-c89f-4563-9090-ce7e7bc65a0d",  
  "title": "New Title",  
  "code": "Updated code goes here",  
  "description": "Updated description",  
  "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",  
  "created_at": "2024-11-03T05:05:24.489Z",  
  "updated_at": "2024-11-03T16:43:31.136Z"  
}
```

- Endpoint: /api/blog

- Description: As a visitor I want to browse through all blogs posted with filters ie. title, tags, content, with pagination
- Allowed Methods: GET
- Payloads:

- Example Request:

<http://localhost:3000/api/blog?page=1&limit=10>

- Example Response:

```
{
  "posts": [
    {
      "id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
      "title": "Updated Blog Post Title",
      "description": "Updated description of the blog post.",
      "content": "Sample content for the blog post.",
      "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
      "created_at": "2024-11-03T16:21:51.931Z",
      "updated_at": "2024-11-03T16:22:29.480Z",
      "numOfUpvotes": 0,
      "numOfDownvotes": 0,
      "numOfReports": 0,
      "author": {
        "username": "tester"
      },
      "tags": [
        {
          "id": "15be47dd-0151-4120-9f0f-b2f3cb8ce567",
          "name": "Tag2"
        },
        {
          "id": "26618c94-78d0-49d0-9875-797bd3540a2a",
          "name": "Tag1"
        },
        {
          "id": "b0e0c1e3-5683-4555-bec5-fcdc094f925b",
          "name": "UpdatedTag1"
        },
        {
          "id": "b3b78415-1251-4611-b893-9fb70ec156f4",
          "name": "UpdatedTag2"
        }
      ],
      "templates": [],
      "comments": [],
      "ratings": []
    }
  ],
  "pagination": {
    "totalPosts": 1,
  }
}
```

```
    "totalPages": 1,  
    "currentPage": 1,  
    "limit": 10  
  }  
}
```

- Endpoint: /api/commenting
 - Description: As a user I want to comment on blog posts or reply to other users
 - Allowed Methods: POST
 - Payloads:

```
blog_post_id : uuid (string)  
content: string  
parent_id : uuid (string)
```

- Example Request: Commenting under Blog Post

```
{  
  "blog_post_id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",  
  "content": "This is a comment unde."  
}
```

- Example Response:

```
{  
  "id": "a0be1569-cb06-44a5-ae8a-c28a0fee76ba",  
  "blog_post_id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",  
  "user_id": "396be59d-df34-4e7f-8226-14ac11d207da",  
  "content": "This is a comment unde.",  
  "parent_id": null,  
  "created_at": "2024-11-03T16:55:14.293Z",  
  "numOfUpvotes": 0,  
  "numOfDownvotes": 0,  
  "numOfReports": 0  
}
```

- Example Request: Replying to a comment

```
{
  "blog_post_id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "content": "This is a reply to a comment.",
  "parent_id": "a0be1569-cb06-44a5-ae8a-c28a0fee76ba"
}
```

- Example Response:

```
{
  "id": "49fa37d6-ee38-4c39-bd97-6c964f1cb329",
  "blog_post_id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "user_id": "396be59d-df34-4e7f-8226-14ac11d207da",
  "content": "This is a reply to a comment.",
  "parent_id": "a0be1569-cb06-44a5-ae8a-c28a0fee76ba",
  "created_at": "2024-11-03T16:57:30.309Z",
  "numOfUpvotes": 0,
  "numOfDownvotes": 0,
  "numOfReports": 0
}
```

- Endpoint: /api/flagging

- Description: As a user I want to be able to report blog posts or comments with an explanation as to why
- Allowed Methods: POST
- Payloads:

```
explanation: string
reporterId: uuid (string)
postId: uuid (string)
```

- Example Request: Under Blog post

```
{
  "explanation": "This post contains inappropriate content.",
  "reporterId": "396be59d-df34-4e7f-8226-14ac11d207da", // Replace with
a valid reporter ID
  "postId": "4f816104-13fe-41b4-bfcf-e4798e845c3a" // Replace with
```



```
a valid post ID
}
```

- Example Response:

```
{
  "id": "b56b45bd-44ec-47d6-84c7-204c98aeba8c",
  "explanation": "This post contains inappropriate content.",
  "createdAt": "2024-11-03T17:01:28.466Z",
  "reporterId": "396be59d-df34-4e7f-8226-14ac11d207da",
  "postId": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "commentId": null
}
```

- Endpoint: /api/rating

- Description: As a user I want to rate comments or blog posts with upvotes (1) or downvotes (-1)
- Allowed Methods: POST
- Payloads:

```
rating: int
targetId: uuid (string)
targetType: string
```

- Example Request:

```
{
  "rating": 1,
  "targetId": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "targetType": "blogPost" // or "comment"
}
```

- Example Response:

```
{
  "id": "ba3a1bf1-c2cf-4ca5-8270-7be5da926fde",
  "rating": 1,
  "user_id": "396be59d-df34-4e7f-8226-14ac11d207da",
}
```

```
{
  "blog_post_id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
  "comment_id": null,
  "targetType": "blogPost"
}
```

- Endpoint: /api/sortedBlog
 - Description: As a visitor I want to see posted blog posts with the highest # of upvotes first and on that blogPost the first comment should have the highest # of upvotes, With Pagination
 - Allowed Methods: GET
 - Payloads:

- Example Request:

```
http://localhost:3000/api/sortedBlog
```

- Example Response:

```
{
  "posts": [
    {
      "id": "4f816104-13fe-41b4-bfcf-e4798e845c3a",
      "title": "Updated Blog Post Title",
      "description": "Updated description of the blog post.",
      "content": "Sample content for the blog post.",
      "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
      "created_at": "2024-11-03T16:21:51.931Z",
      "updated_at": "2024-11-03T17:04:54.406Z",
      "numOfUpvotes": 1,
      "numOfDownvotes": 0,
      "numOfReports": 1,
      "author": {
        "username": "tester"
      },
      "tags": [
        {
          "id": "15be47dd-0151-4120-9f0f-b2f3cb8ce567",
          "name": "Tag2"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "id": "26618c94-78d0-49d0-9875-797bd3540a2a",
      "name": "Tag1"
    },
    {
      "id": "b0e0c1e3-5683-4555-bec5-fcdc094f925b",
      "name": "UpdatedTag1"
    },
    {
      "id": "b3b78415-1251-4611-b893-9fb70ec156f4",
      "name": "UpdatedTag2"
    }
  ],
  "comments": [
    {
      "id": "a0be1569-cb06-44a5-ae8a-c28a0fee76ba",
      "blog_post_id":
"4f816104-13fe-41b4-bfcf-e4798e845c3a",
      "user_id": "396be59d-df34-4e7f-8226-14ac11d207da",
      "content": "This is a comment unde.",
      "parent_id": null,
      "created_at": "2024-11-03T16:55:14.293Z",
      "numOfUpvotes": 0,
      "numOfDownvotes": 0,
      "numOfReports": 0,
      "user": {
        "username": "tester",
        "avatar": "/uploads/avatars/avatar1.png"
      },
      "ratings": []
    },
    {
      "id": "49fa37d6-ee38-4c39-bd97-6c964f1cb329",
      "blog_post_id":
"4f816104-13fe-41b4-bfcf-e4798e845c3a",
      "user_id": "396be59d-df34-4e7f-8226-14ac11d207da",
      "content": "This is a reply to a comment.",
      "parent_id": "a0be1569-cb06-44a5-ae8a-c28a0fee76ba",
      "created_at": "2024-11-03T16:57:30.309Z",
      "numOfUpvotes": 0,
```

```
        "numOfDownvotes": 0,
        "numOfReports": 0,
        "user": {
            "username": "tester",
            "avatar": "/uploads/avatars/avatar1.png"
        },
        "ratings": []
    }
],
"ratings": [
    {
        "id": "ba3a1bf1-c2cf-4ca5-8270-7be5da926fde",
        "rating": 1,
        "user_id": "396be59d-df34-4e7f-8226-14ac11d207da",
        "blog_post_id":
"4f816104-13fe-41b4-bfcf-e4798e845c3a",
        "comment_id": null,
        "targetType": "blogPost"
    }
]
},
{
    "id": "f9da97eb-1fba-4ce2-8410-814ded7402d5",
    "title": "Sample Blog Pwost244",
    "description": "This is a test blog post.",
    "content": "Sample content for the blog post.",
    "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
    "created_at": "2024-11-03T17:07:11.497Z",
    "updated_at": "2024-11-03T17:07:11.497Z",
    "numOfUpvotes": 0,
    "numOfDownvotes": 0,
    "numOfReports": 0,
    "author": {
        "username": "tester"
    },
    "tags": [
        {
            "id": "15be47dd-0151-4120-9f0f-b2f3cb8ce567",
            "name": "Tag2"
        },
        {
```

```
        "id": "26618c94-78d0-49d0-9875-797bd3540a2a",
        "name": "Tag1"
      }
    ],
    "comments": [],
    "ratings": []
  }
},
{
  "pagination": {
    "totalPosts": 2,
    "totalPages": 1,
    "currentPage": 1,
    "limit": 10
  }
}
```

- Endpoint: /api/tempInBlog
 - Description: As a visitor, I want to see the list of blog posts that mention a code template on the template page. With pagination
 - Allowed Methods: GET
 - Payloads:

- Example Request:

http://localhost:3000/api/tempInBlog

- Example Response:

```
[
  {
    "id": "721ce7d8-4b97-495b-b4dc-297c341d0a99",
    "title": "Sample Blog Pwost244",
    "description": "This is a test blog post.",
    "content": "Sample content for the blog post.",
    "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
    "created_at": "2024-11-03T17:10:51.299Z",
    "updated_at": "2024-11-03T17:10:51.299Z",
    "numOfUpvotes": 0,
```

```
    "numOfDownvotes": 0,
    "numOfReports": 0,
    "templates": [
      {
        "id": "77b2e88d-9b27-4622-b3f6-3fe121f2a73e",
        "title": "bwwrosj",
        "code": "console.log('Hello World');",
        "description": "This template prints Hello World to the
console.",
        "author_id": "396be59d-df34-4e7f-8226-14ac11d207da",
        "created_at": "2024-11-03T16:43:25.544Z",
        "updated_at": "2024-11-03T16:43:25.544Z"
      }
    ],
    "author": {
      "id": "396be59d-df34-4e7f-8226-14ac11d207da",
      "first_name": "test",
      "last_name": "testL",
      "username": "tester",
      "email": "tester@gmail.com",
      "password":
"$2a$10$06eClwBMWlujdDOOX13lY.dTkxiegQTcIoayWlKYgDshY.Zf2ZexK",
      "avatar": "/uploads/avatars/avatar1.png",
      "phone_number": "1111111111",
      "role": "USER",
      "createdAt": "2024-11-03T05:05:04.495Z",
      "updatedAt": "2024-11-03T05:05:04.495Z"
    },
    "tags": [
      {
        "id": "15be47dd-0151-4120-9f0f-b2f3cb8ce567",
        "name": "Tag2"
      },
      {
        "id": "26618c94-78d0-49d0-9875-797bd3540a2a",
        "name": "Tag1"
      }
    ]
  }
]
```

- Endpoint: /api/execution

- Description:

As a visitor (unauthenticated user), I want to write code in various programming languages (e.g., C, C++, Java, Python, JavaScript) on Scriptorium.

As a visitor, I want to provide input to my code via the standard input (stdin) before execution so that I can test programs that require user input.

- Allowed Methods: POST

- Payloads:

```
{
  "Language": "Python"
  "Code": "name = input('Enter Name:')\nprint(f'Hello, {name}!')"
  "stdin": "World"
}
```

- Example Request:

```
http://localhost:3000/api/execution
```

- Example Response:

```
[
  {
    "Output": "Enter name: Hello, World!\n",
    "Error": ""
  }
]
```