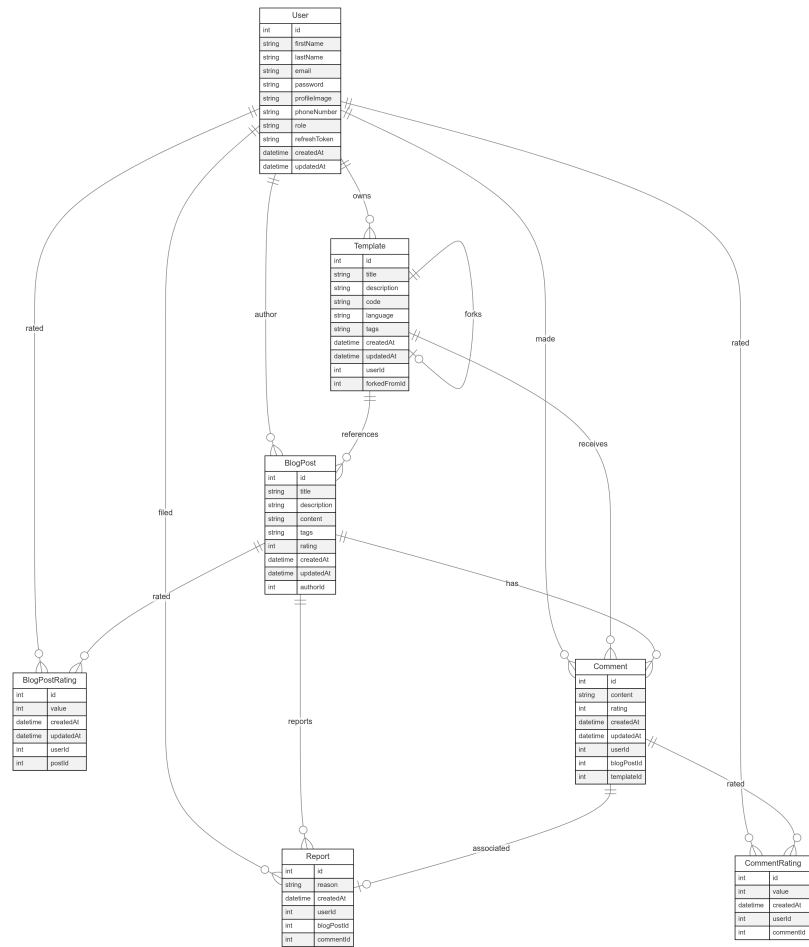


API Documentation — CSC309 PP1

Damian Melia & Rami Al-Azzawe

Model Diagram



User Profile

Get User Profile

- **Method:** GET
- **URL:** /api/user/profile
- **Description:** Fetches details of the authenticated user's profile.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Response:**
 - 200 OK: { **user:** { id, firstName, lastName, email, phoneNumber, profileImage, createdAt, updatedAt } }
 - 404 Not Found: { **error:** "User not found" }
 - 500 Internal Server Error: { **error:** "Failed to fetch user details" }

Update User Profile

- **Method:** PATCH
- **URL:** /api/user/profile
- **Description:** Updates the authenticated user's profile.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Form Data:**
 - **firstName:** String
 - **lastName:** String
 - **email:** String
 - **phoneNumber:** String
 - **password:** String (hashed)
 - **profileImage:** File (optional)
- **Response:**
 - 200 OK: { **message:** "Profile updated successfully", user }
 - 400 Bad Request: { **error:** "Failed to update profile" }
 - 500 Internal Server Error: { **error:** "Failed to upload image" }

Code Execution

Execute Code

- **Method:** POST
- **URL:** /api/code/execute
- **Description:** Executes code in a specified language and returns output or errors.
- **Body:**
 - **code:** String (code to execute)
 - **language:** String (e.g., "python", "javascript", "c", "cpp")
 - **stdin:** String (optional input for the code)
- **Response:**
 - 200 OK: { **output:** "execution result" }
 - 400 Bad Request: { **error:** "Execution error" }
 - 500 Internal Server Error: { **error:** "Error message" }

Templates

Get Templates

- **Method:** GET
- **URL:** /api/templates
- **Description:** Retrieves a list of templates. Supports pagination and filtering by title, tags, and content.
- **Query Parameters:**
 - **title:** String (filter by title)
 - **tags:** String (comma-separated tags)
 - **content:** String (search within content)
 - **page:** Integer (pagination page number)
 - **limit:** Integer (pagination limit)
 - **ownedByUser:** Boolean (set to true to retrieve only the user's templates)
- **Response:**
 - 200 OK: { **templates:** [...], **pagination:** { **totalItems**, **totalPages**, **page**, **limit** } }
 - 500 Internal Server Error: { **error:** "Failed to fetch templates" }

Create Template

- **Method:** POST
- **URL:** /api/templates
- **Description:** Creates a new template for the authenticated user.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Body:**
 - **title:** String
 - **description:** String
 - **code:** String
 - **language:** String
 - **tags:** String (comma-separated)
- **Response:**
 - 201 Created: { **template** }
 - 400 Bad Request: { **error:** "Failed to create template" }

Fork Template

- **Method:** POST
- **URL:** /api/templates/fork
- **Description:** Forks an existing template to create a modified version.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Body:**
 - **id:** Integer (ID of the original template)
 - **title, description, tags, code:** Optional fields for customization
- **Response:**
 - 201 Created: { **forkedTemplate** }
 - 404 Not Found: { **error:** "Original template not found" }

Authentication

Login

- **Method:** POST
- **URL:** /api/auth/login
- **Description:** Logs in a user with email and password, setting authentication cookies.
- **Body:**
 - email: String
 - password: String
- **Response:**
 - 200 OK: { **message:** "Login successful", **accessToken**, **refreshToken** }
 - 401 Unauthorized: { **error:** "Invalid email or password" }

Logout

- **Method:** POST
- **URL:** /api/auth/logout
- **Description:** Logs out the authenticated user and clears authentication cookies.
- **Headers:**
 - Authorization: Bearer token (access token)
- **Response:**
 - 200 OK: { **message:** "Logout successful" }
 - 500 Internal Server Error: { **error:** "Failed to logout" }

Posts

Create Post

- **Method:** POST
- **URL:** /api/posts
- **Description:** Creates a new blog post by an authenticated user.

- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Body:**
 - **title:** String (required) - The title of the blog post
 - **description:** String (required) - Short description of the post
 - **content:** String (required) - Main content of the post
 - **tags:** String (optional, comma-separated) - Tags for the post
- **Response:**
 - 201 Created: { **message:** "Post created successfully", **post:** Post object }
 - 400 Bad Request: { **error:** "Failed to create post" }

Get Posts

- **Method:** GET
- **URL:** /api/posts
- **Description:** Retrieves a list of blog posts, with optional pagination and filtering.
- **Query Parameters:**
 - **tags:** String (optional, comma-separated) - Filter posts by tags
 - **page:** Integer (optional) - Pagination page number
 - **limit:** Integer (optional) - Number of posts per page
- **Response:**
 - 200 OK: { **posts:** [...], **pagination:** { **totalItems**, **totalPages**, **page**, **limit** } }
 - 500 Internal Server Error: { **error:** "Failed to fetch posts" }

Update Post

- **Method:** PUT
- **URL:** /api/posts/{id}
- **Description:** Updates an existing blog post by the post's author.
- **Headers:**
 - **Authorization:** Bearer token (access token)

- **Body:**
 - **title:** String (optional) - Updated title
 - **description:** String (optional) - Updated description
 - **content:** String (optional) - Updated content
 - **tags:** String (optional, comma-separated) - Updated tags
- **Response:**
 - 200 OK: { **message:** "Post updated successfully", **post:** Updated post object }
 - 403 Forbidden: { **error:** "Unauthorized to edit this post" }
 - 404 Not Found: { **error:** "Post not found" }

Delete Post

- **Method:** DELETE
- **URL:** /api/posts/{id}
- **Description:** Deletes a blog post by the post's author.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Response:**
 - 204 No Content: Post successfully deleted
 - 403 Forbidden: { **error:** "Unauthorized to delete this post" }
 - 404 Not Found: { **error:** "Post not found" }

Comments

Create Comment

- **Method:** POST
- **URL:** /api/posts/{postId}/comments
- **Description:** Adds a comment to a specific blog post.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Body:**
 - **content:** String (required) - Content of the comment

- **Response:**
 - 201 Created: { **message:** "Comment created successfully", **comment:** Comment object }
 - 400 Bad Request: { **error:** "Failed to create comment" }

Get Comments

- **Method:** GET
- **URL:** /api/posts/{postId}/comments
- **Description:** Retrieves comments for a specific blog post, with pagination.
- **Query Parameters:**
 - **page:** Integer (optional) - Pagination page number
 - **limit:** Integer (optional) - Number of comments per page
- **Response:**
 - 200 OK: { **comments:** [...], **pagination:** { **totalItems,** **totalPages,** **page,** **limit** } }
 - 500 Internal Server Error: { **error:** "Failed to fetch comments" }

Update Comment

- **Method:** PUT
- **URL:** /api/posts/{postId}/comments/{id}
- **Description:** Updates an existing comment by the comment's author.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Body:**
 - **content:** String (optional) - Updated content of the comment
- **Response:**
 - 200 OK: { **message:** "Comment updated successfully", **comment:** Updated comment object }
 - 403 Forbidden: { **error:** "Unauthorized to edit this comment" }
 - 404 Not Found: { **error:** "Comment not found" }

Delete Comment

- **Method:** DELETE
- **URL:** /api/posts/{postId}/comments/{id}
- **Description:** Deletes a comment by the comment's author.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Response:**
 - 204 No Content: Comment successfully deleted
 - 403 Forbidden: { **error:** "Unauthorized to delete this comment" }
 - 404 Not Found: { **error:** "Comment not found" }

Reports

Report a Blog Post

- **Method:** POST
- **URL:** /api/posts/{postId}/report
- **Description:** Allows an authenticated user to report a blog post.
- **Headers:**
 - **Authorization:** Bearer token (access token)
- **Body:**
 - **reason:** String (required) - Reason for reporting the post
- **Response:**
 - 201 Created: { **message:** "Report submitted successfully", **report:** Report object }
 - 400 Bad Request: { **error:** "Failed to submit report" }

Report a Comment

- **Method:** POST
- **URL:** /api/posts/{postId}/comments/{commentId}/report
- **Description:** Allows an authenticated user to report a comment.
- **Headers:**

- **Authorization:** Bearer token (access token)
- **Body:**
 - **reason:** String (required) - Reason for reporting the comment
- **Response:**
 - 201 Created: { **message:** "Report submitted successfully", **report:** Report object }
 - 400 Bad Request: { **error:** "Failed to submit report" }

Get Reports for a Blog Post

- **Method:** GET
- **URL:** /api/posts/{postId}/reports
- **Description:** Retrieves all reports submitted for a specific blog post. (Admin only)
- **Headers:**
 - **Authorization:** Bearer token (admin access token)
- **Response:**
 - 200 OK: { **reports:** [...] }
 - 403 Forbidden: { **error:** "Unauthorized access" }

Get Reports for a Comment

- **Method:** GET
- **URL:** /api/posts/{postId}/comments/{commentId}/reports
- **Description:** Retrieves all reports submitted for a specific comment. (Admin only)
- **Headers:**
 - **Authorization:** Bearer token (admin access token)
- **Response:**
 - 200 OK: { **reports:** [...] }
 - 403 Forbidden: { **error:** "Unauthorized access" }

Delete a Report

- **Method:** DELETE
- **URL:** /api/reports/{reportId}
- **Description:** Deletes a specific report by ID. (Admin only)
- **Headers:**
 - **Authorization:** Bearer token (admin access token)
- **Response:**
 - 204 No Content: Report successfully deleted
 - 403 Forbidden: { **error:** "Unauthorized access" }
 - 404 Not Found: { **error:** "Report not found" }