Task Management API Documentation

Overview

The Task Management API has been developed using Django and Django REST Framework, providing a robust platform for performing CRUD operations on tasks. The API offers a range of endpoints, including authentication, task creation, retrieval, updating, and deletion. Authentication is handled securely using JSON Web Tokens (JWT), ensuring that sensitive operations are accessible only to authorized users.

The API's core features include:

1. **Authentication Endpoint**:
   - Obtain a JWT token by providing valid credentials (username and password).
2. **Authenticated Endpoint:**
   - A secure endpoint requiring a valid JWT token for authentication, offering an extra layer of security.
3. **Task Endpoints**:
   - Retrieve a list of tasks, create new tasks, update existing tasks, and delete tasks by ID.

Authentication Endpoint:

- POST /api/token/

(Obtain a JWT token by providing a valid username and password.)

Authenticated Endpoint:

- GET /api/authenticated/

(A secure endpoint that requires a valid JWT token for authentication.)

Task Endpoints:

- GET /api/task-list/

  :(Retrieve a list of all tasks.)

- POST /api/task-create/

  :(Create a new task. Requires authentication.)

- GET /api/task-update/{id}/

  :(Retrieve details of a specific task by ID.)

- POST /api/task-update/{id}/

  :(Update details of a specific task by ID. Requires authentication.)

- DELETE /api/task-delete/{id}/

  :(Delete a specific task by ID. Requires authentication.)

Authentication in Postman

1. Obtain Token

- Open Postman and create a new request.
- Set the request type to POST.
- Enter the authentication endpoint URL: *http://api-url/api/token/*.
- Go to the 'Body' tab.
- Select 'x-www-form-urlencoded'.
- Add two key-value pairs: username and password with your actual credentials [admin,admin]**.**
- Click on Send.
- In the response, you will get a JSON object containing the access token.
- Copy the access token from the response.

Endpoints:

Authenticated Endpoint in Postman

1. Create a New Request

- Open Postman and create a new request.

2. Set Request Details

- Set the request type to GET.
- Enter the authenticated endpoint URL: http://api-url/api/authenticated/ (replace http://api-url with your actual API URL).

3. Add Authorization Header

- Go to the Headers tab.
- Add a new header:
    - Key: Authorization
    - Value: Bearer <your_access_token> (replace <your_access_token> with your actual access token).

4. Send the Request

- Click on **Send**.

5. Verify Response

- In the response, you should see a status code of **200** (OK).
- The response body should contain: **{"message": "You are authenticated!"}**.

By following these steps, you can test an authenticated request to the **/api/authenticated/** endpoint in Postman. Make sure to replace **http://api-url** and **<your_access_token>** with your actual API URL and access token.

Task Endpoints in Postman

1. Retrieve List of Tasks

- Create a new request in Postman.
- Set the request type to GET.
- Enter the task list endpoint URL: http://api-url/api/task-list/ (replace http://api-url with your actual API URL).

2. Create a New Task

- Create another new request in Postman.
- Set the request type to POST.
- Enter the create task endpoint URL: http://api-url/api/task-create/ (replace http://api-url with your actual API URL).
- Go to the Headers tab and add the Authorization header with the value Bearer <your_access_token>.
- Go to the Body tab and provide the JSON data with task details (title, body, status).

3. Update a Task

- Create a new request in Postman.
- Set the request type to POST.
- Enter the update task endpoint URL: http://api-url/api/task-update/{id}/ (replace {id} with the actual task ID).
- Go to the Headers tab and add the Authorization header with the value Bearer <your_access_token>.
- Go to the Body tab and provide the JSON data with updated task details.

4. Delete a Task

- Create a new request in Postman.
- Set the request type to DELETE.
- Enter the delete task endpoint URL: http://api-url/api/task-delete/{id}/ (replace {id} with the actual task ID).
- Go to the Headers tab and add the Authorization header with the value Bearer <your_access_token>.

5. Send Requests and Verify Responses

- Click on Send for each request.
- Verify that the responses match the expected status codes and bodies as described in the documentation below.

Error Handling

Status Code 401 (Unauthorized):

- Invalid or missing JWT token.

Status Code 403 (Forbidden):

- Insufficient permissions.

Status Code 404 (Not Found):

- Resource not found.

Status Code 500 (Internal Server Error):

- Unexpected server error.

Example:

Example 1: Creating a Task:

- Create a new request in Postman.
- Set the request type to POST.
- Enter the create task endpoint URL: http://127.0.0.1:8000/api/task-create/.
- Go to the Headers tab and add the Authorization header with the value Bearer <your_access_token>.
- Go to the Body tab and provide the JSON data with task details (title, body, status).
- Click on Send

Example 2: Updating a Task:

- Create a new request in Postman.
- Set the request type to POST.
- Enter the update task endpoint URL: http://127.0.0.1:8000/api/task-update/1/ (replace 1 with the actual task ID).
- Go to the Headers tab and add the Authorization header with the value Bearer <your_access_token>.

- Go to the Body tab and provide the JSON data with updated task details.
- Click on Send.

Example 3: Deleting a Task:

- Create a new request in Postman.
- Set the request type to DELETE.
- Enter the delete task endpoint URL: http://127.0.0.1:8000/api/task-delete/1/ (replace 1 with the actual task ID).
- Go to the Headers tab and add the Authorization header with the value Bearer <your_access_token>.
- Click on Send.

By using these examples, you can test the create, update, and delete functionality of your API using Postman. Make sure to replace http://api-url, <your_access_token>, and 1 with your actual API URL, access token, and task ID.

As a foundational component for **task management**, the API sets the stage for building applications that require efficient and secure task handling. I have tried to create well-documented endpoints and clear authentication process to make it a developer-friendly solution for a variety of projects. Ongoing maintenance and further feature development can be seamlessly integrated, ensuring the longevity and adaptability of the Task Management API .