

Secure Task Manager API

Postman Collection Documentation

Prepared By: Khushi Raghav

This document contains the API testing collection created using Postman for a Secure Task Manager backend project.

User Registration API

User Registration API: This endpoint allows new users to register by providing username, email, and password. It validates existing users and ensures unique registration.

The screenshot shows the Secure Task Manager API interface. On the left, the sidebar lists collections, environments, and flows. Under the 'Secure Task Manager API' collection, the 'POST Register User' endpoint is selected. The request URL is `http://localhost:5000/api/auth/register`. The 'Body' tab is active, showing the following JSON payload:

```
1 {
2   "username": "khushi",
3   "email": "khushi454@gmail.com",
4   "password": "123456"
5 }
```

Below the request, the response is displayed. The status bar indicates a **400 Bad Request** response with a duration of 7 ms and a size of 277 B. The response body is a JSON object:

```
1 {
2   "message": "User already exists"
3 }
```

User Login API

User Login API: This endpoint authenticates registered users using email and password. Upon successful login, a JWT token is generated for secure access.

The screenshot shows the Secure Task Manager API interface. On the left, the sidebar lists collections, environments, and flows. Under the 'Secure Task Manager API' collection, the 'POST Login User' endpoint is selected. The main panel shows a POST request to `http://localhost:5000/api/auth/login`. The request body is set to raw and contains the following JSON:

```
1 {  
2   "email": "khush545@gmail.com",  
3   "password": "123456"  
4 }  
5
```

Below the request, the response is shown as a 200 OK status with a response time of 270 ms and a size of 419 B. The response body is a JSON object containing a token:

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZC16IjY50GEWZjU0ZmNiMTNkZDYxMmIwMDJkMSIsImIhdCI6MTc3MDY1OTM1MiwiZhwIjoxNzcnNjYyOTUyfQ.  
Ukej-X_axghitFlRfiNgCcQE1Z25eMYnqk0PVu3ZpE"  
3 }
```

Create Task API

Create Task API: This endpoint allows authenticated users to create a new task. The task is linked to the logged-in user.

The screenshot shows the Secure Task Manager API interface. On the left, the sidebar lists collections, environments, history, flows, and a search bar. Under the 'Secure Task Manager API' section, the 'POST Create Task' endpoint is selected. The main panel shows a POST request to `http://localhost:5000/api/tasks`. The 'Body' tab is active, displaying the JSON payload:

```
1 [
2   "title": "Learn Secure Task Manager"
3 ]
```

Below the request, the response is shown in a JSON format:

```
1 {
2   "title": "Learn Secure Task Manager",
3   "user": "698a0f54fcfb13d612b002d1",
4   "_id": "698a10b6ebcfdb890663ea40b",
5   "_v": 0
}
```

The status bar at the bottom indicates a `201 Created` response with `88 ms`, `353 B`, and a `Save Response` button.

Get My Tasks API

Get My Tasks API: This endpoint retrieves all tasks associated with the authenticated user using the provided token.

The screenshot shows the 'Secure Task Manager API / Get My Tasks' endpoint in the Secure Task Manager application. The left sidebar lists various collections and environments, with 'Secure Task Manager API' expanded to show endpoints like 'Register User', 'Login User', 'Create Task', and 'Get My Tasks'. The 'Get My Tasks' endpoint is selected and highlighted with an orange border. The main panel displays the request configuration: method 'GET', URL 'http://localhost:5000/api/tasks', and a body section indicating 'none'. Below the request details, the response is shown as a JSON array containing one task object:

```
[{"_id": "698a10b6ebcf890663ea40b", "title": "Learn Secure Task Manager", "user": "698a0f54fcbb13dd612b002d1", "__v": 0}]
```

Update Task API

Update Task API: This endpoint allows users to update an existing task's details securely using its unique ID.

The screenshot shows the Secure Task Manager API interface. On the left sidebar, under the 'Secure Task Manager API' section, the 'PUT UPDATE TASK' endpoint is selected. The main panel displays the API request configuration. The method is set to 'PUT' and the URL is 'http://localhost:5000/api/tasks/698a10b6ebcf890663ea40b'. The 'Body' tab is active, showing the JSON payload:

```
1 {
2   "title": "Updated task title"
3 }
```

Below the request, the response is shown in a JSON format:

```
1 {
2   "_id": "698a10b6ebcf890663ea40b",
3   "title": "Updated task title",
4   "user": "698a0f54fcbb13dd612b002d1",
5   "__v": 0
6 }
```

Delete Task API

Delete Task API: This endpoint deletes a specific task using its ID. It ensures that only authorized users can remove their tasks.

The screenshot shows the Secure Task Manager API interface in a web browser. The left sidebar lists collections, environments, history, flows, and the current workspace, "Khushi RagHAV's Workspace". Under "Secure Task Manager API", the "Delete Task" endpoint is selected. The main panel displays the DELETE request URL: `http://localhost:5000/api/tasks/698a10b6ebcf890663ea40b`. The "Body" tab is selected, showing the response body: `{ "message": "Task deleted successfully" }`. Other tabs include "Cookies", "Headers (8)", "Test Results", and "Save Response". The status bar at the bottom indicates a 200 OK response with 60 ms latency and 274 B size.

Environment Variables Setup

Environment Variables Setup: This section shows configured environment variables such as `baseUrl` and `token` to manage dynamic request handling.

The screenshot shows a workspace interface with the following components:

- Left Sidebar:** Includes sections for "Collections", "Environments" (with "Secure Task Manager" selected), "History", and "Flows".
- Top Bar:** Shows tabs like "New", "Import", "Secure", "CONFLIK", "POST Log", "POST Creat", "Secur", "Flows", "GET Get", "PUT UPDA", and "Secure Task Manager".
- Central Area:** A table titled "Secure Task Manager" showing environment variables:

Variable	Value
baseUrl	http://localhost:5000
token	eyJhbGciOiJIUzI1NiJ9.eyJrZXkiOiJpZC16YSGFwZjU0ZmNlMTNz... [REDACTED]
- Right Panel:** Titled "Fixing the request" for a "POST Login User" task.
 - Request Details:** Shows "Fix this request" and "Review 2 Changes" buttons.
 - HTTP Request Editor:** Contains fields for "Edit HTTP request", "Allow agent to update this request?", "URL" (set to "http://localhost:5000/api/auth/register"), and "Body (raw)" (set to "http://localhost:5000/api/auth/login").

Headers Configuration

Headers Configuration: This demonstrates setting Content-Type and Authorization headers to ensure secure API communication.

Headers			9 hidden
Key	Value	Description	Bulk Edit Presets
<input checked="" type="checkbox"/> Content-Type	application/json		
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...		
Key	Value	Description	

Authorization Setup

Authorization Setup: This section shows Bearer Token authentication using environment variables for secure access control.

The screenshot shows the Postman interface with the following details:

- Header Bar:** Docs, Params, **Authorization**, Headers (11), Body, Scripts, Settings. The "Authorization" tab is selected.
- Authorization Type:** Bearer
- Token:** {{token}}
- Description:** The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.