📘 TeamHub API Documentation

Base URL: `http://localhost:3000/`

## 🔐 Auth & User Module

### Create a new user account.

**Method: POST**

**Endpoint: /auth/signup**

**Body:**

**{**

**"name": "John Doe",**

**"email": "john@example.com",**

**"password": "securePass123"**

**}**

**cURL:**

**curl -X POST http://localhost:3000/auth/signup \**

**-H "Content-Type: application/json" \**

**-d '{ "name": "John Doe", "email": "john@example.com", "password": "securePass123" }'**

### Login and receive JWT token

**Method: POST**

**EndPoint: /auth/login**

**Body:**

**{**

**"email": "john@example.com",**

**"password": "securePass123"**

**}**

**cURL:**

**curl -X POST http://localhost:3000/auth/login \**

**-H "Content-Type: application/json" \**

**-d '{ "email": "john@example.com", "password": "securePass123" }'**

### Get authenticated user info

**Method: GET**

**Endpoint: /auth/getuser**

**Header: Authorization: Bearer <JWT\_TOKEN>**

**cURL:**

**curl -X GET http://localhost:3000/auth/getuser \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

## 👥 Team Module

### Create a new team

**Method: POST**

**Endpoint: /teams/create**

**Header:**

**Authorization: Bearer <JWT\_TOKEN>**

**Body:**

**{**

**"name": "Team Alpha"**

**}**

**cURL:**

**curl -X POST http://localhost:3000/teams/create \**

**-H "Authorization: Bearer <JWT\_TOKEN>" \**

**-H "Content-Type: application/json" \**

**-d '{ "name": "Team Alpha" }'**

### Invite user to a team by email (Mock Only)

**Method: POST**

**Endpoint: /teams/send-invite/:id**

**Body:**

**{**

**"email": "invitee@example.com"**

**}**

**cURL:**

**curl -X POST http://localhost:3000/teams/send-invite/<TEAM\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>" \**

**-H "Content-Type: application/json" \**

**-d '{ "email": "invitee@example.com" }'**

### Get all pending invites

**Method: GET**

**Endpoint: /teams/my-invites**

**cURL:**

**curl -X GET http://localhost:3000/teams/my-invites \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Accept a pending invite

**Method: POST**

**Endpoint: /teams/accept-invite/:inviteId**

**cURL:**

**curl -X POST http://localhost:3000/teams/accept-invite/<INVITE\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Reject a pending invite

**Method: POST**

**Endpoint: /teams/reject-invite/:inviteId**

**cURL:**

**curl -X POST http://localhost:3000/teams/reject-invite/<INVITE\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Get all team members of a Team

**Method: GET**

**Endpoint: /teams/get-team-members/:id**

**cURL:**

**curl -X GET http://localhost:3000/teams/get-team-members/<TEAM\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

## 📁 Project Module

### Create a project in a team

**Method: POST  
Endpoint: /projects/create**

**Body:**

**{**

**"name": "Project Mercury",**

**"description": "Project for Q3 launch",**

**"teamId": "<TEAM\_ID>" //optional**

**}**

**cURL:**

**curl -X POST http://localhost:3000/projects/create \**

**-H "Authorization: Bearer <JWT\_TOKEN>" \**

**-H "Content-Type: application/json" \**

**-d '{ "name": "Project Mercury", "description": "Project for Q3 launch", "teamId": "<TEAM\_ID>" }'**

### Get My projects

**Method: GET**

**Endpoint: /projects/my-projects?page=1&limit=10**

**Query Params:**

**Page, limit**

**cURL:**

**curl -X GET http://localhost:3000/projects/my-projects \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Get projects by team ID

**Method: GET**

**Endpoint: /projects/team/:teamId**

**cURL:**

**curl -X GET http://localhost:3000/projects/team/<TEAM\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Get project by ID

**Method: GET**

**Endpoint: /projects/:id**

**cURL:**

**curl -X GET http://localhost:3000/projects/<PROJECT\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Update project details

**Method: PUT**

**Endpoint: /projects/:id**

**Body:**

**{**

**"name": "Updated Project",**

**"description": "Updated description"**

**}**

**cURL:**

**curl -X PUT http://localhost:3000/projects/<PROJECT\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>" \**

**-H "Content-Type: application/json" \**

**-d '{ "name": "Updated Project", "description": "Updated description" }'**

### Delete project if all tasks are DONE

**Method: DELETE**

**Endpoint: /projects/:id**

**cURL:**

**curl -X DELETE http://localhost:3000/projects/<PROJECT\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

## ✅ Task Module

### Create a new task

**Method: POST**

**Endpoint: /tasks/create**

**Body:**

**{**

**"title": "Design UI",**

**"description": "Initial UI",**

**"dueDate": "2025-07-01",**

**"projectId": "<PROJECT\_ID>",**

**"assigneeId": "" //optional**

**}**

**cURL:**

**curl -X POST http://localhost:3000/tasks/create \**

**-H "Authorization: Bearer <JWT\_TOKEN>" \**

**-H "Content-Type: application/json" \**

**-d '{ "title": "Design UI", "description": "Initial UI", "dueDate": "2025-07-01", "projectId": "<PROJECT\_ID>" }'**

### Add comment to task

**Method: POST**

**Endpoint: /tasks/comment/:taskId**

**Body:**

**{**

**"comment": "Completed frontend logic"**

**}**

**cURL:**

**curl -X POST http://localhost:3000/tasks/comment/<TASK\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>" \**

**-H "Content-Type: application/json" \**

**-d '{ "comment": "Completed frontend logic" }'**

### List all tasks with filters

**Method: GET**

**Endpoint: /tasks/all**

**Query Params:**

**title, status, assignee, createdBy**

**cURL:**

**curl -X GET "http://localhost:3000/tasks/all?status=TODO" \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Get task by History

**Method: GET  
Endpoint: /tasks/history/:taskId**

**cURL:**

**curl -X GET http://localhost:3000/tasks/<TASK\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

### Update Task

**Method: PUT  
Endpoint: /tasks/:taskId**

**Body:**

**{**

**"title": "Updated Task Title",**

**"description": "Updated description",**

**"dueDate": "2025-07-01T12:00:00.000Z",**

**"assigneeId": "", //optional**

**"status": "INPROGRESS"**

**}**

**cURL:**

**curl -X GET http://localhost:3000/tasks/<TASK\_ID> \**

**-H "Authorization: Bearer <JWT\_TOKEN>"**

## 🔑 Token Handling

**All protected routes require this header:**

**Authorization: Bearer <JWT\_TOKEN>**

**NOTE:**

**If the token is expired:**

* **Re-login to generate a new token**
* **Handle 401 Unauthorized on the frontend**