**COSC 310**

# API Documentation

Khizar Aamir, Abdullah Naqvi, Abdur Rehman, Haider Ali, Abdullah Munir

This document describes the Task Management System API endpoints, authentication requirements, and request formats for developers integrating with the backend.

## Base URL

http://localhost:3200

## Authentication

The API uses JWT (JSON Web Token) based authentication:

• ***Access To*ken**: Include in Authorization header for API requests

• ***Refresh Token***: Stored as HTTP-only cookie for obtaining new access tokens

#### Authentication Header

For protected endpoints, include:

Authorization: Bearer <access_token>

#### User Roles

• ***User***: Regular access to own tasks and data

• ***Admin***: Full system access

## Core API Endpoints

### Authentication

#### Login

• ***URL***: /auth

• ***Method***: POST

• ***Request Body***:

```
{
  "email": "user@example.com",
  "password": "password123"
}
```

• ***Response***: Access token and user information

**Refresh Token**

• ***URL***: /auth/refresh

• ***Method***: GET

• ***Response***: New access token

**Logout**

• ***URL***: /auth/logout

• ***Method***: POST

## User Endpoints

**Get User Profile**

• ***URL***: /user

• ***Method***: GET

• ***Response***: User profile data

**Update User Profile**

• ***URL***: /user

• ***Method***: PATCH

• ***Request Body***: Fields to update (name, phone)

**Search Users**

• ***URL***: /user/search?q=search_term

• ***Method***: GET

• ***Response***: Matching user profiles

## Task Endpoints

**Get User Tasks**

• ***URL***: /tasks

• ***Method***: GET

• ***Response***: List of tasks assigned to user

**Create Task**

• ***URL***: /tasks

• ***Method***: POST

• ***Request Body***:

```
{
 "title": "Task title",
 "description": "Task description",
 "priority": "Medium",
 "dueDate": "2023-12-15T00:00:00.000Z",
 "assignees": ["userId1", "userId2"]
}
```

**Get Task Details**

• ***URL***: /tasks/:taskId

• ***Method***: GET

**Update Task**

• ***URL***: /tasks/:taskId

• ***Method***: PATCH

• **Request Body**: Fields to update (status, priority, etc.)

**Delete Task**

• ***URL***: /tasks/:taskId

• ***Method***: DELETE

**Task Assignee Management**

• ***Add Assignee***: PATCH /tasks/:taskId/assignees

• ***Remove Assignee***: DELETE /tasks/:taskId/assignees

## Admin Endpoints

### (Require admin role)

**User Management**

• ***Get All Users***: GET /admin/users

• ***Invite User***: POST /admin/users

• ***Update User***: PATCH /admin/users/:userId

• ***Delete User***: DELETE /admin/users/:userId

**Task Management (Admin)**

• ***Get All Tasks***: GET /admin/tasks

• ***Lock Task***: PATCH /admin/tasks/:taskId/lock

• ***Unlock Task****:* PATCH /admin/tasks/:taskId/unlock

**Admin Analytics**

• ***Get Metrics***: GET /admin/metrics

## Conversation Endpoints

**Messaging**

• ***Get Conversations***: GET /conversations

• ***Create Conversation***: POST /conversations

• ***Get Messages***: GET /conversations/:conversationId/messages

• ***Send Message***: POST /conversations/:conversationId/messages

## Sample Request

```
curl -X POST http://localhost:3200/auth \
  -H "Content-Type: application/json" \
  -d '{"email": "user@example.com", "password": "password123"}'
```

## Error Responses

The API uses standard HTTP status codes and returns error messages in a consistent format:

```
{
  "message": "Error description"
}
```

**Common error codes**:

• ***400***: Bad Request (invalid input)

• ***401***: Unauthorized (authentication required)

• ***403***: Forbidden (insufficient permissions)

• ***404***: Not Found

• ***500***: Internal Server Error

For validation errors, the response includes details about each invalid field.

**Real-time Features**

The application uses **Socket.io** for real-time communication, enabling:

• Live chat messages

• Task assignment notifications

• Task update notifications

Socket connections require the same JWT authentication used for the REST API.