

# task-manager-api

The Task Manager API allows users to manage tasks. It includes endpoints for user registration, login, and task management (CRUD operations). Tasks can be filtered by their completion status.

## users

The **Users** endpoints handle user authentication, including registration and login requests. After a successful login, a JSON Web Token (JWT) is issued for authenticating future requests. This token should be included in the `Authorization` header, using the Bearer scheme, for access to protected resources in the API.

### POST user-register

`http://localhost:3000/api/users/register`

Creates a new user account.

**Body** raw (json)

json

```
{
  "name": "ahmed",
  "email": "ahmed@gmail.com",
  "password": "12345"
}
```

### POST user-login

`http://localhost:3000/api/users/login`

Authenticates a user and returns a JWT token.

## Body raw (json)

---

json

```
{
  "email": "doaa@gmail.com",
  "password": "123"
}
```

---

## tasks

The **Tasks** endpoints allow users to manage their to-do items, enabling them to create, read, update, delete, and filter tasks. Each user has access only to their own tasks, which requires authentication through a JSON Web Token (JWT).

---

### GET get-all-tasks

http://localhost:3000/api/tasks

Retrieves a list of all tasks for the authenticated user.

#### HEADERS

---

Authorization

Bearer

---

### POST create-task

http://localhost:3000/api/tasks/

Adds a new task to the user's task list. Each task has properties like title, description, and status (e.g., completed or incomplete).

#### HEADERS

---

Authorization

Bearer

## Body raw (json)

---

json

```
{
  "title": "new task",
  "completed": false
}
```

---

## PATCH update-task

http://localhost:3000/api/tasks?\_id=672e59256603d37271016e7a

Updates the details of a specific task, such as marking it as completed.

## HEADERS

---

Authorization

Bearer

## PARAMS

---

\_id

672e59256603d37271016e7a

## Body raw (json)

---

json

```
{
  "title": "updated task",
  "completed": false
}
```

---

## DELETE delete-task

http://localhost:3000/api/tasks?\_id=672e563a872b1dfbf759746d

Removes a specific task from the user's task list.

## HEADERS

---

Authorization	Bearer
---------------	--------

## PARAMS

---

_id	672e563a872b1dfbf759746d
-----	--------------------------

---

## GET get-single-task

<http://localhost:3000/api/tasks/672d115a40f5ee8612338792>

Retreive a single task by id.

## HEADERS

---

Authorization	Bearer
---------------	--------

## GET filter-tasks

<http://localhost:3000/api/tasks/filter?completed=false>

Filters the user's tasks based on their status, such as completed or incomplete.

## HEADERS

---

Authorization	Bearer
---------------	--------

## PARAMS

---

completed	false
-----------	-------