

## TASK MANAGEMENT API

### Login:

Login to access the task apis

### API :

Authentication Required : No

Method type : POST

URL : http://localhost:5000/login

Header : { Content-type: application/json }

Description : This API is used to validate the user is authenticated or not.

### Request Payload:

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

### Response:

#### (1). IF the user is Valid

```
{
  "message": "Login successful",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImVtYWlsIjoibWFuaUBnbWFpbC5jb20iLCJpYXQiOiJlMzZMzQwMTA5MTMsImV4cCI6MTczNDAxNDUxM30.wh9IttMVt8IFdR-6cltnjhnX77HsNC8Jt4rLTtvU2VQ",
  "user": {
    "id": 1,
    "username": "mani",
    "email": "mani@gmail.com"
  }
}
```

#### (2). IF the user is not Valid

```
{
  "message": "Invalid credentials"
}
```

**Register API:**

Register is used to create a new user record.

**API :**

Authentication Required : No

Method type : POST

URL : http://localhost:5000/register

Header : { Content-type: application/json }

Description : This API is used to create a new user. And a password is encrypt and store to the database

**Payload:**

```
{  
  "username": "manis",  
  "email" : "manis@gmail.com",  
  "password": "mani1"  
}
```

**Response:****(1). IF User is valid**

```
{  
  "message": "User created successfully",  
  "user": {  
    "id": 2,  
    "username": "manis",  
    "email": "manis@gmail.com"  
  }  
}
```

**(2).Example : User is not valid using same email address**

```
{  
  "message": "Email already in use."  
}
```

## 1. Task Creation :

Create the New Task

### API :

Authentication Required : yes

Method type : POST

URL : <http://localhost:5000/createtasks>

Header : {Authorization: Bearer <token>, Content-type: application/json}

Description : This API create the tasks

### Payload:

```
{
  "title": "task1",
  "description": "task1 creation",
  "dueDate": "2024-12-13",
  "status": "pending"
}
```

### Response:

```
{
  "id": 1,
  "title": "task1",
  "description": "task1 creation",
  "dueDate": "2024-12-13T00:00:00.000Z",
  "status": "pending",
  "userId": 1,
  "updatedAt": "2024-12-12T09:43:57.702Z",
  "createdAt": "2024-12-12T09:43:57.702Z"
}
```

## 2.GET TASK:

Get All Task data or filtered data using status and dueDate

### API:

Authentication Required : yes

Method type : GET

URL : <http://localhost:5000/tasks>

Header : {Authorization: Bearer <token>, Content-type: application/json}

Description : This API gets all the tasks

**Response:**

```
[
  {
    "id": 1,
    "title": "task1",
    "description": "task1 creation",
    "dueDate": "2024-12-13T00:00:00.000Z",
    "status": "Pending",
    "userId": 1,
    "createdAt": "2024-12-12T09:43:57.000Z",
    "updatedAt": "2024-12-12T09:43:57.000Z"
  },
  {
    "id": 2,
    "title": "task2",
    "description": "task2 creation",
    "dueDate": "2024-12-13T00:00:00.000Z",
    "status": "Pending",
    "userId": 1,
    "createdAt": "2024-12-12T10:18:32.000Z",
    "updatedAt": "2024-12-12T10:18:32.000Z"
  }
]
```

**(OR)**

**Request Payload:**

Filter data using dueDate or status (Completed, Pending, In Progress)

```
{
  "status": "Completed",
  "dueDate": "2024-12-15"
}
```

**3. UPDATE TASK:****API:**

Authentication Required : yes

Method type : PUT

URL : <http://localhost:5000/updatetasks/1>

Header : {Authorization: Bearer <token>, Content-type: application/json}

Description : This API to update task informations

**Payload :**

Need to updated data send on payload

Example:

```
{  
  "status": "Completed"  
}
```

**Response:**

```
{  
  "message": "Task Updated Successfully",  
  "data": {  
    "id": 1,  
    "title": "task1",  
    "description": "task1 creation",  
    "dueDate": "2024-12-13T00:00:00.000Z",  
    "status": "Completed",  
    "userId": 1,  
    "createdAt": "2024-12-12T09:43:57.000Z",  
    "updatedAt": "2024-12-12T13:06:38.000Z"  
  }  
}
```

**4. Delete Task:**

Delete particular Task data using id

**API:**

Authentication Required: yes

Method type: DELETE

URL : <http://localhost:5000/deletetasks/1>

Header : {Authorization: Bearer <token>, Content-type: application/json}

Description : This API to delete the task data

**Response:**

```
{  
  "message": "Task deleted successfully"  
}
```

## 5. Get Count :

Get count of the each status with start date and end date and time filter

### API:

Authentication Required : yes

Method type : GET

URL : <http://localhost:5000/getCount/1>

Header : {Authorization: Bearer <token>, Content-type: application/json}

Description : This API is used to get the count data depend on the payload

### Payload:

```
{  
    "startDate": "2024-12-10",  
    "endDate": "2024-12-16"  
}  
(or)  
{  
    "startDate": "2024-12-10",  
    "endDate": "2024-12-16"  
    "status": "Pending"  
}
```

### Response:

```
{  
    "count": 2  
}
```