

# API Documentation - Task Management API

## Overview:

This API allows users to manage a list of tasks with basic CRUD (Create, Read, Update, Delete) operations. Tasks are stored in memory and each task has an ID, a text description, a completion status, and a timestamp.

**Base URL:** <http://localhost:5000>

## 1. GET /tasks

- Description: Retrieve all tasks.
- Method: GET
- URL: /tasks
- Response:
  1. Status Code: 200 OK
  2. Body:

```
[
  {
    "id": 1,
    "text": "Learn API",
    "completed": false,
    "time": "2024-12-30 14:00:00"
  },
  {
    "id": 2,
    "text": "Write documentation",
    "completed": true,
    "time": "2024-12-30 14:30:00"
  }
]
```

## 2. GET /tasks/:id

- Description: Retrieve a specific task by ID.
- Method: GET
- URL: /tasks/:id (Replace :id with the task ID)
- Example Request: /tasks/1
- Response:
  1. Status Code: 200 OK
  2. Body:

```
{
  "id": 1,
  "text": "Learn API",
  "completed": false,
  "time": "2024-12-30 14:00:00"
}
```
- Error Response:

1.Status Code: 404 Not Found

2.Body:

```
{  
  "message": "Task not found"  
}
```

### 3. POST /tasks

- **Description:** Create a new task.
- **Method:** POST
- **URL:** /tasks
- **Request Body:**

```
{  
  "text": "Buy groceries"  
}
```

- **Response:**

1.Status Code: 201 Created

2.Body:

```
{ "id": 3, "text": "Buy groceries", "completed": false, "time": "2024-12-30 14:40:00" }
```

- **Error Response:**

If text is missing:

```
{ "message": "Text is required" }
```

### 4. PUT /tasks/:id

- **Description:** Update an existing task by ID.
- **Method:** PUT
- **URL:** /tasks/:id
- **Request Body:**  
{ "text": "Complete documentation", "completed": true }

- **Response:**

1.Status Code: 200 OK

2.Body:

```
{ "id": 1, "text": "Complete documentation", "completed": true, "time": "2024-12-30 15:00:00" }
```

- **Error Response:**

```
{ "message": "Task not found" }
```

## 5. DELETE /tasks/:id

- **Description:** Delete a task by ID.
- **Method:** DELETE
- **URL:** /tasks/:id
- **Example Request:** /tasks/1
- **Response:**
  - Status Code: 204 No Content
  - Body: Empty response (task deleted successfully)
- **Error Response:**
  - If task not found:

```
{ "message": "Task not found" }
```

### Error Handling:

- **400 Bad Request:** Missing or invalid data in the request body.
  - Example:{"message": "Text is required"}
- **404 Not Found:** When a task is not found (e.g., when trying to fetch or delete a task that doesn't exist).
  - Example:{"message": "Task not found"}
- **500 Internal Server Error:** Unexpected errors that occur during the execution of the server.
  - Example:{ "message": "Internal server error"}

### Summary:

- This API supports CRUD operations for managing tasks. Tasks are stored temporarily in memory and can be manipulated via the provided endpoints. All responses are in JSON format, and errors are handled with appropriate HTTP status codes and messages.

