Task Management API Documentation

- ## Base URL

```
http://localhost:5000
```

- ## Endpoints

1. Create Task

Endpoint: `/tasks`
Method: `POST`
Description: Creates a new task.

Request Body:

```
json

{
    "Title": "Task Title",
    "Description": "Task Description",
    "Status": "Not Started"
}
```

Response:

- **201 Created**

  ```
  json

  {
      "message": "Task created"
  }
  ```

- **400 Bad Request**: If required fields are missing or invalid.

---

2. Read Tasks

Endpoint: `/tasks`
Method: `GET`
Description: Retrieves a list of all tasks.

Response:

- **200 OK**

  ```
  json

   [
      {
          "Task_id": 1,
          "Title": "Task Title",
          "Description": "Task Description",
          "Status": "Not Started"
      },
      ...
  ]
  ```

- **500 Internal Server Error**: If there is an error retrieving the tasks.

---

3. Update Task

Endpoint: `/tasks/<task_id>`
Method: `PUT`
Description: Updates the details of an existing task.

Request Body:

```
json

{
    "Title": "Updated Task Title",
    "Description": "Updated Task Description",
    "Status": "In Progress"
}
```

Response:

- **200 OK**

  ```
  json

  {
      "message": "Task updated"
  }
  ```

- **404 Not Found**: If the task ID does not exist.
- **400 Bad Request**: If required fields are missing or invalid.

---

4. Delete Task

Endpoint: `/tasks/<task_id>`
Method: `DELETE`
Description: Deletes an existing task.

Response:

- **200 OK**

  ```
  json

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

- **404 Not Found**: If the task ID does not exist.

Error Handling

All endpoints may return the following error responses:

- **400 Bad Request**: Indicates that the server cannot or will not process the request due to a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

  ```
  json

  {
      "message": "Bad Request"
  }
  ```

- **404 Not Found**: Indicates that the server can't find the requested resource. This status code is returned if the requested resource does not exist.

  ```
  json

  {
      "message": "Not Found"
  }
  ```

- **500 Internal Server Error**: Indicates that the server encountered an unexpected condition that prevented it from fulfilling the request.

  ```
  json

  {
      "message": "Internal Server Error"
  }
  ```

Example cURL Commands

Create Task

```
curl -X POST http://localhost:5000/tasks -H "Content-Type:
application/json" -d '{"Title": "New Task", "Description": "This is a new
task", "Status": "Not Started"}''
```

Read Tasks

```
curl -X GET http://localhost:5000/tasks
```

Update Task

```
curl -X PUT http://localhost:5000/tasks/1 -H "Content-Type:
application/json" -d '{"Title": "Updated Task", "Description": "This is an
updated task", "Status": "In Progress"}'
```

Delete Task

```
curl -X DELETE http://localhost:5000/tasks/1
```

Notes

- Ensure your server is running before making requests to these endpoints.
- Replace `localhost:5000` with your server’s actual base URL if it's different.
- Proper error handling should be implemented in the client application to handle various responses from these endpoints.