## **Task Management API Documentation**

### **Introduction**

The Task Management API is a simple Flask application that allows users to manage tasks. It provides endpoints for retrieving, creating, updating, and deleting tasks.

### **Installation**

* Clone the repository:

**git clone <repository\_url>**

Install the required dependencies:

**pip install flask**

### **Usage**

### Running the Application

Run the Flask application using the following command:

**python app.py**

**The application will start on** http://127.0.0.1:5000/ **by default.**

#### **Endpoints**

* GET /tasks
  + Retrieve a list of tasks.
  + Parameters:
    - sort\_by **(optional): Sort tasks by a specific field (default: 'id').**
    - order **(optional): Sort order ('asc' or 'desc', default: 'asc').**
    - filter\_status **(optional): Filter tasks by status.**
    - page **(optional): Page number for pagination (default: 1).**
    - per\_page **(optional): Number of tasks per page (default: 3).**
* **POST /tasks**
  + **Create a new task.**
  + **Parameters:**
    - Title**: Title of the task.**
    - Description**: Description of the task.**
    - DueDate**: Due date of the task.**
    - Status**: Status of the task ('Pending', 'Completed', etc.).**
* **GET /tasks/{id}**
  + **Retrieve details of a specific task.**
  + **Parameters:**
    - id**: Task ID.**
* **PUT /tasks/{id}**
  + **Update details of a specific task.**
  + **Parameters:**
    - id**: Task ID.**
    - Title**: Updated title.**
    - Description**: Updated description.**
    - DueDate**: Updated due date.**
    - Status**: Updated status.**
* **DELETE /tasks/{id}**
  + **Delete a specific task.**
  + **Parameters:**
    - id**: Task ID.**
* **PUT /tasks/{id}/complete**
  + **Mark a task as completed.**
  + **Parameters:**
    - id**: Task ID.**

### **Testing**

**The application includes unit tests for critical components. Run the tests using the following command:**

**python -m unittest test\_app.py**

### **Conclusion**

**The Task Management API provides a straightforward way to manage tasks through a RESTful interface. For further customization or extension, refer to the Flask documentation and the provided source code.**

**Please note that this is a basic documentation template, and you may want to customize it based on your specific application requirements. Additionally, consider adding more details, such as API request/response examples, security considerations, and error handling information.**