



# TASK MANAGEMENT SYSTEM

REST-API DOCUMENTATION

PRESENTED BY:-

SAI KUMAR GUNDEBOINA

CELL NO: 8639338924

MAIL ID: [GUNDEBOINASAIKUMAR86@GMAIL.COM](mailto:GUNDEBOINASAIKUMAR86@GMAIL.COM)

# About Project

- ▶ This project is implementation of REST APIs for the Task Management System.
- ▶ It includes creation, deletion, update of task.
- ▶ This project is coded in python language using Flask framework.
- ▶ This project provides the Create, Rename, Update and Delete operations of tasks.
- ▶ Each task consists of fields like: id, Title, Description, Due date, Created date, Updated date.
- ▶ This project has basic Authentication(username : admin, password : 1234) to access these APIs.
- ▶ In this project, I used json file("data.json") to store the task data.

# Methods

- ▶ **GET** /tasks : to get all the tasks →
- ▶ **GET** /tasks/{id} : to get the particular task →
- ▶ **POST** /tasks : to create a new task →
- ▶ **PUT** /tasks/{id} : to update the fields of an existing task →
- ▶ **PATCH** /tasks/{id}/complete : to mark a task with that id as complete. →
- ▶ **DELETE** /tasks/{id} : to delete an existing task. →

# GET /tasks

- ▶ This endpoint is used to get all the Tasks.
- ▶ It will return json array of Tasks with the fields id, status, title, description, due\_date, created\_date, updated\_date
- ▶ **Filtering**: we can filter the data using fields like status, due\_date, title.
- ▶ Ex : API to fetch only completed tasks.
  - ▶ **GET** /tasks?status=completed

# GET /tasks/{id}

- ▶ This endpoint is used to get all fields of that particular id
- ▶ It will return json array of Task with the fields id, status, title, description, due\_date, created\_date, updated\_date

# POST /tasks

- ▶ This endpoint is used to create new Task.
- ▶ It requires all the field like id, status, title, description, due\_date.
- ▶ created\_at will be updated by the current datetime.
- ▶ Conditions:
  - ▶ **Id** must be unique.
  - ▶ **due\_date** must be in future.
  - ▶ **status** must be in "pending" or "in\_progress" or "completed".

# PUT /tasks{id}

- ▶ This endpoint is used to update the existing task.
- ▶ It requires all the fields similar to above method.
- ▶ If the task is not exists, then it will create a new task.
- ▶ updated\_at will be updated by the current datetime.

# PATCH /tasks/{id}/complete

- ▶ This endpoint is used to mark a Task as completed
- ▶ It requires only id of that task.

# DELETE /tasks{id}

- ▶ This endpoint is used to delete the existing task.
- ▶ It requires only id of that task.
- ▶ It will returns success on successful removal and returns error message if that task is not available.



# THANK YOU

for your time...