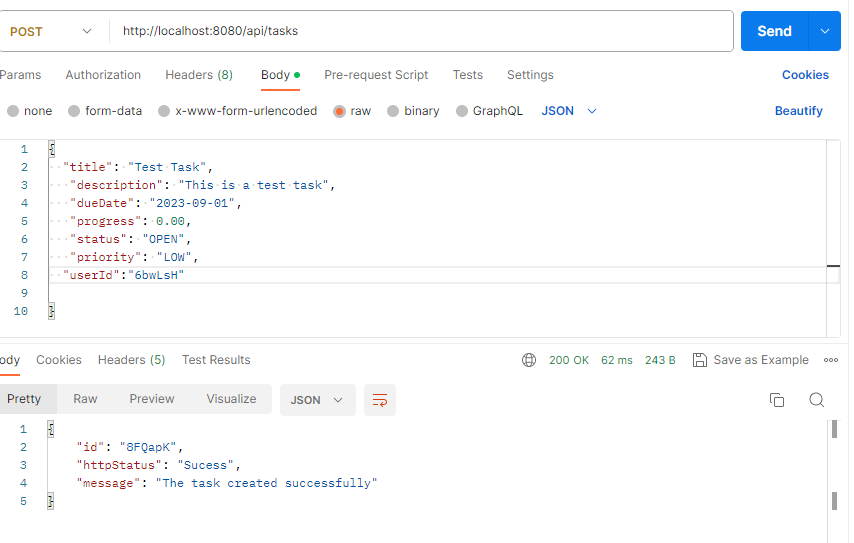
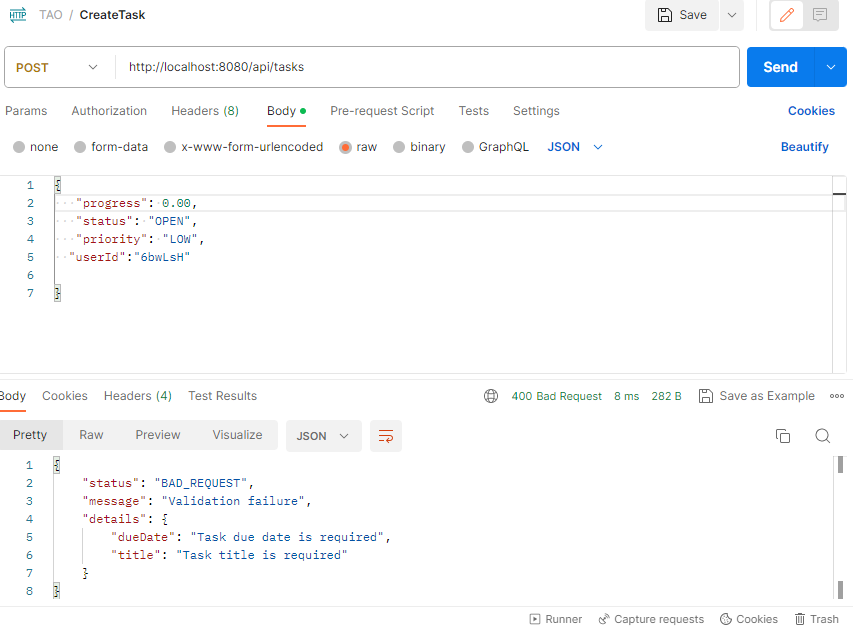
**Tao Assignement:**

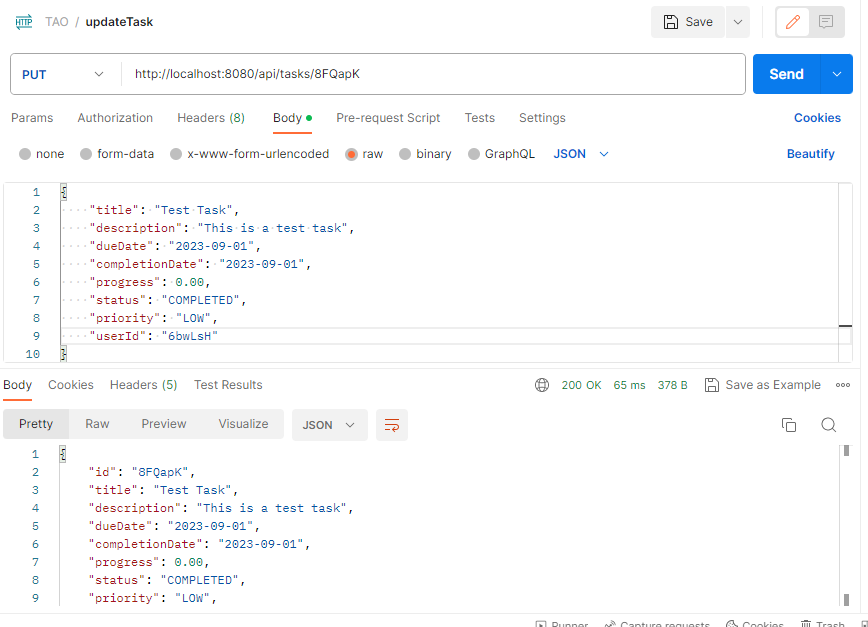
**Create Task: 1. Endpoint: POST /api/tasks**



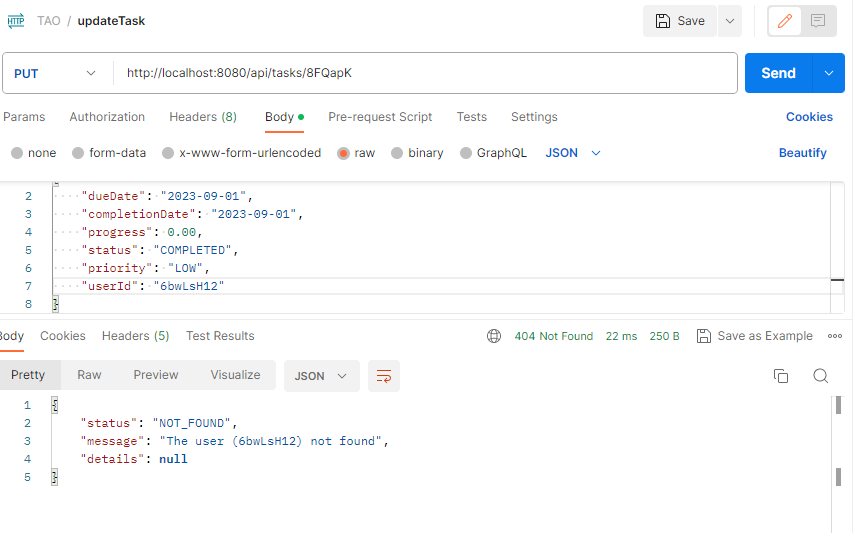
Create task validation for title and duedate (Status and Priority setting default to OPEN and LOW)



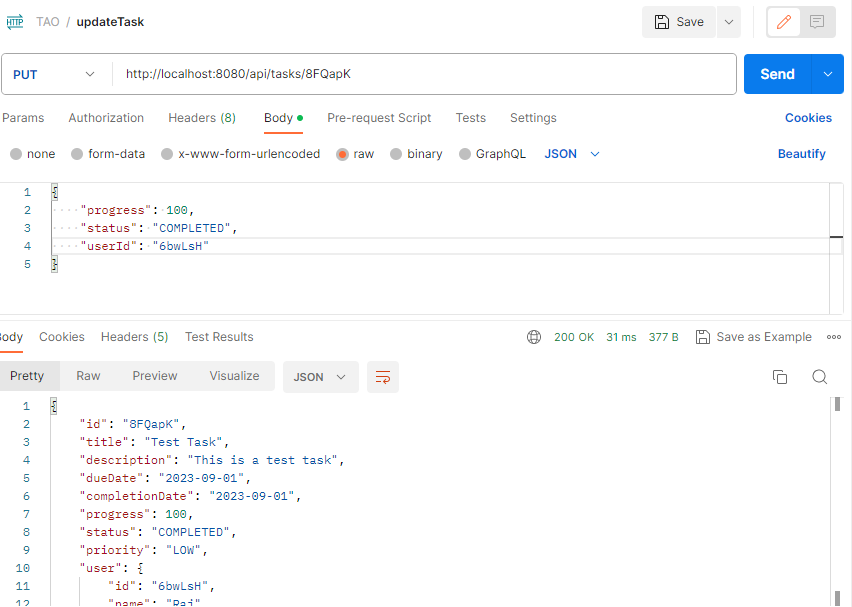
**Update task: 2. Endpoint: PUT /api/tasks/{taskId}**

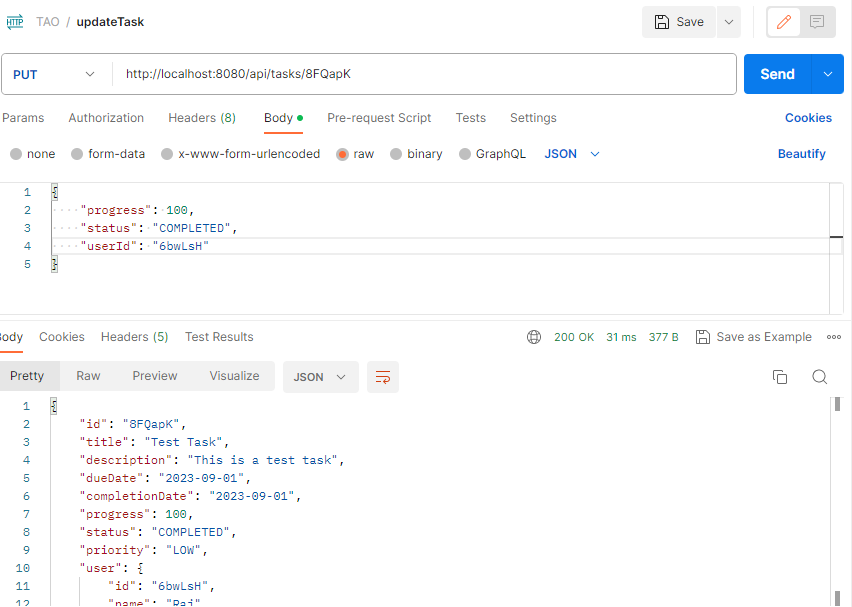


If not passing valid userId validation will fail.

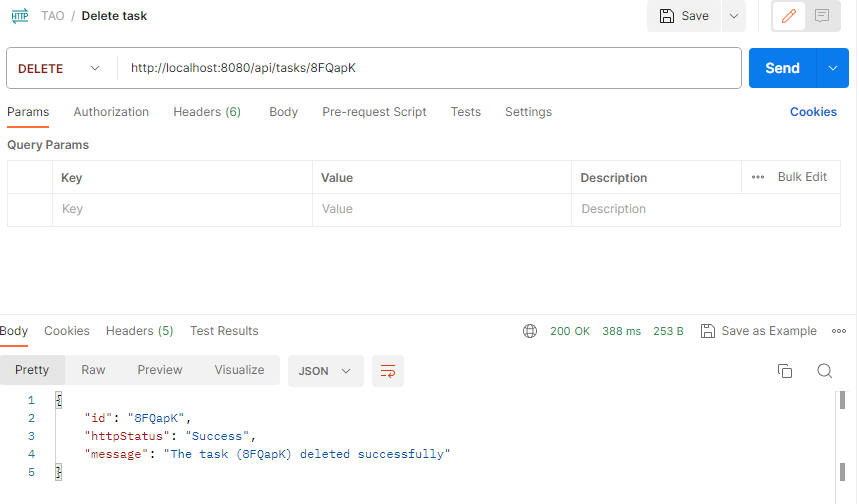


UserId and TaskId is mandatory other fields are option while updating the task.

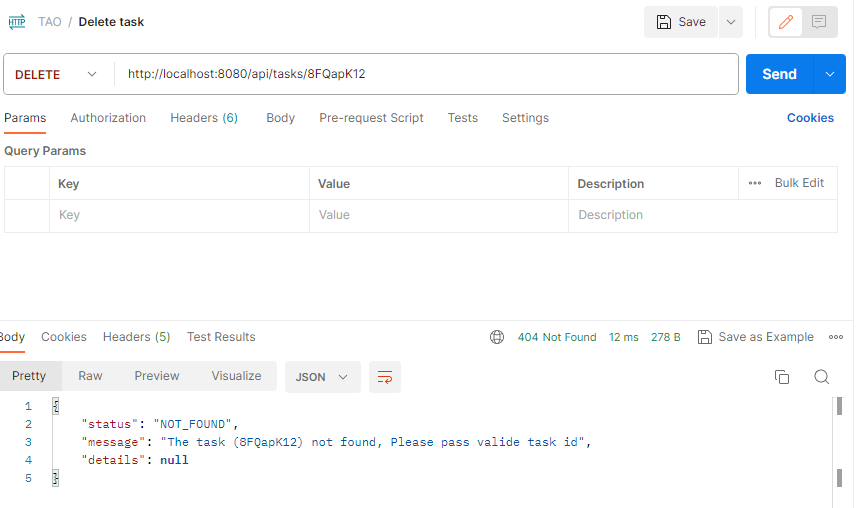




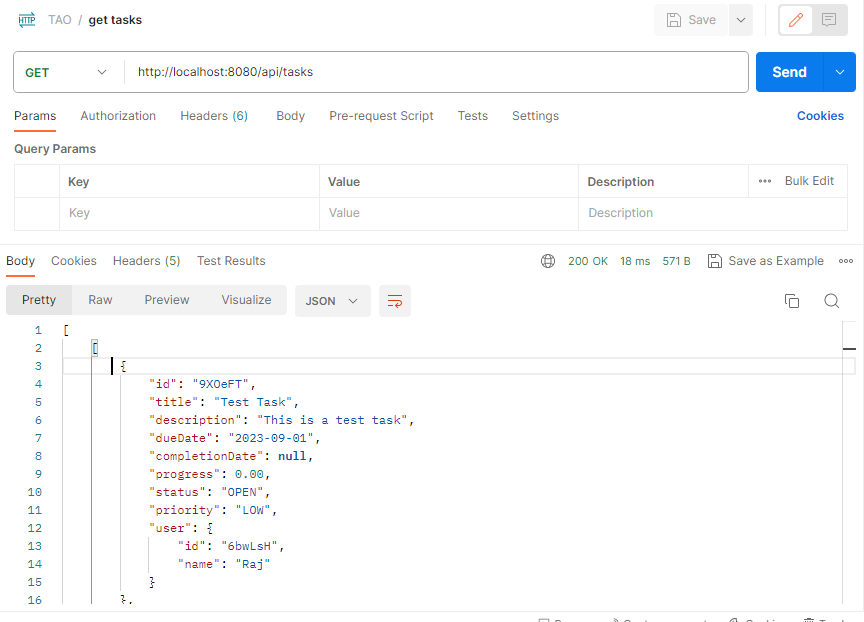
**Delete Task with taskId – 3. Endpoint: DELETE /api/tasks/{taskId}**



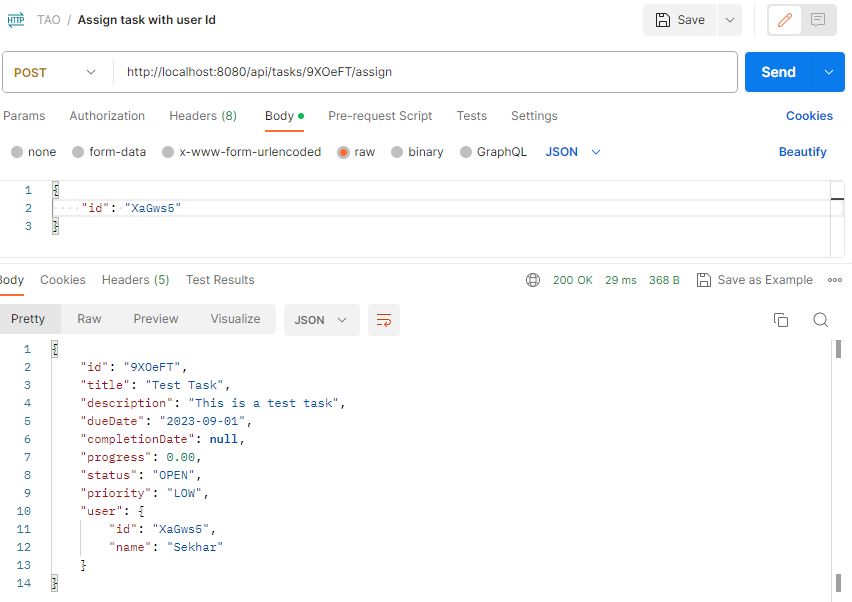
Task Id is mandatory while deleting the task.



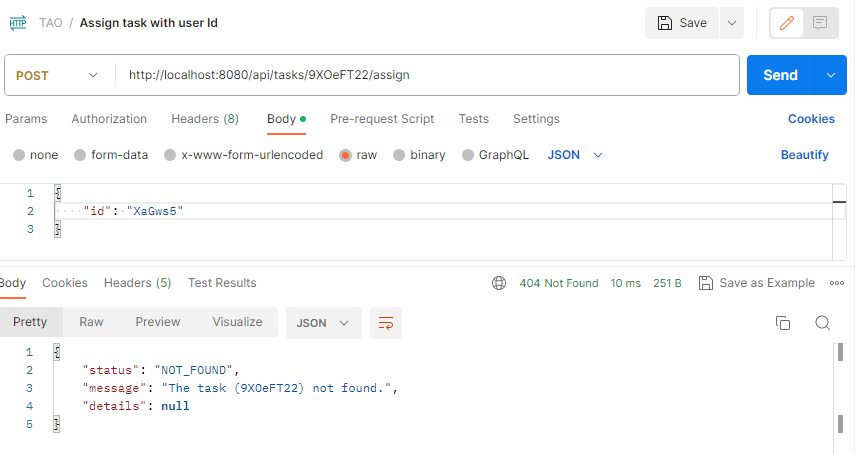
**Get List of tasks: 4. Endpoint: GET /api/tasks – Done**

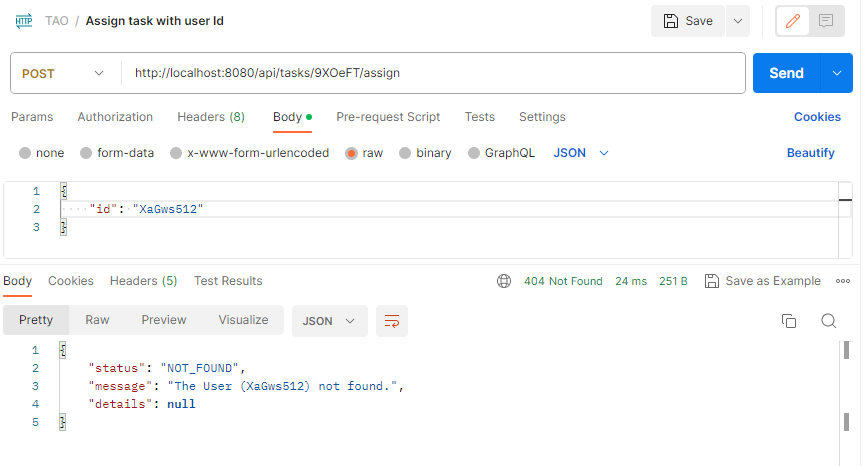


**Assign task to the user: 5. Endpoint: POST /api/tasks/{taskId}/assign**

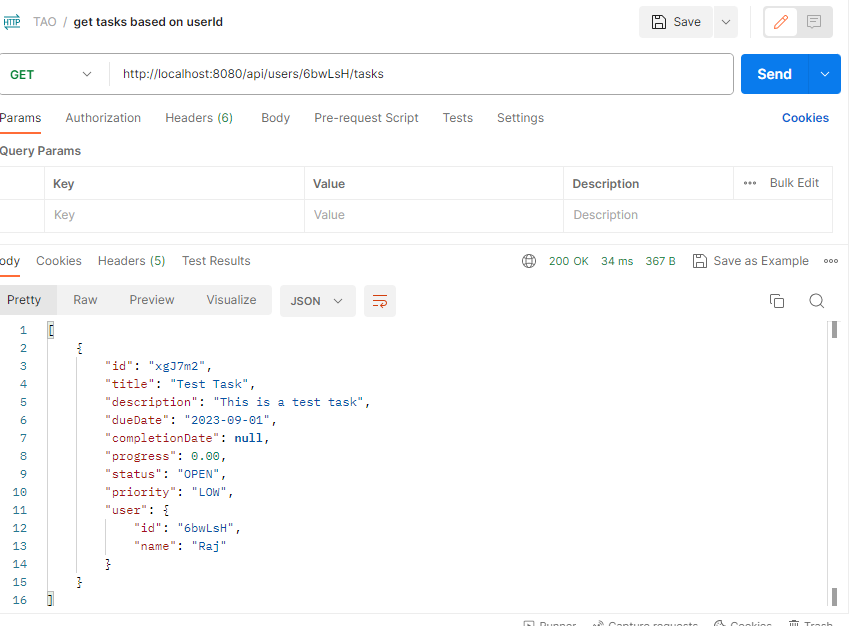


Validation error if task id or user id is not valid:

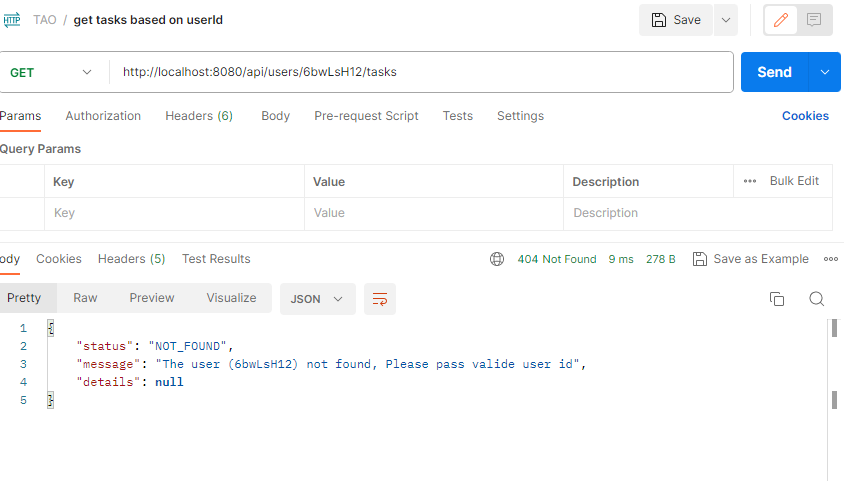




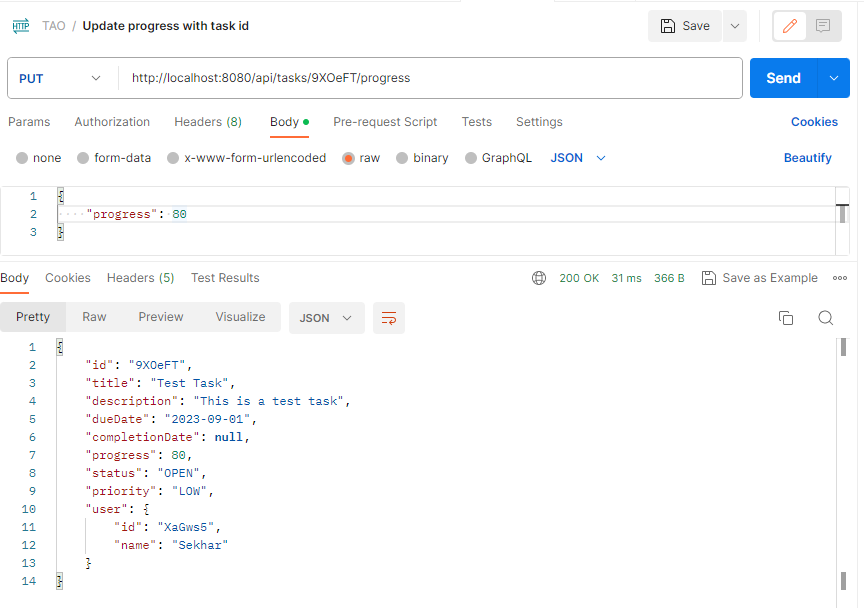
**Get task list based on user Id: 6. Endpoint: GET /api/users/{userId}/tasks**



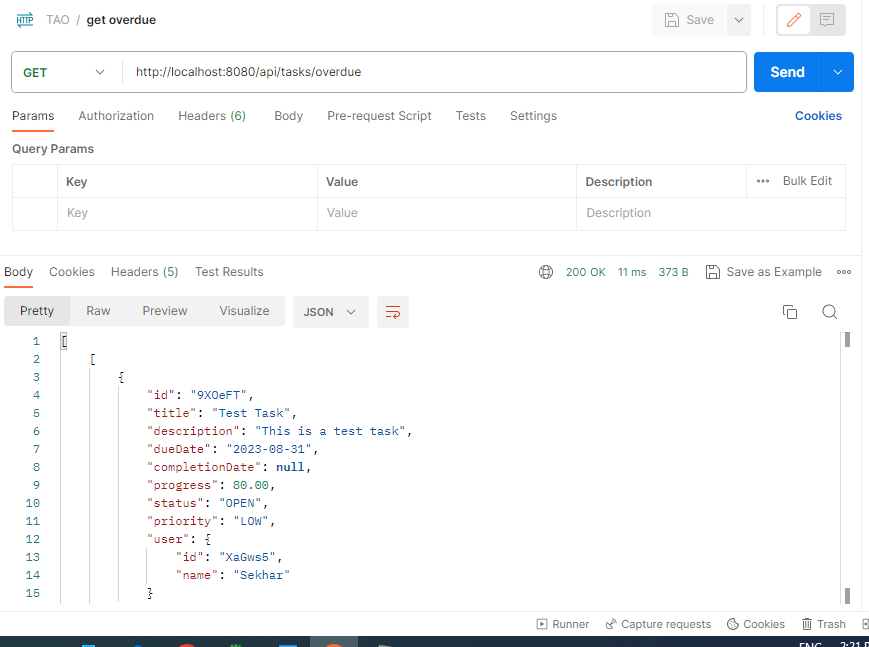
Validation if user id is not valid:



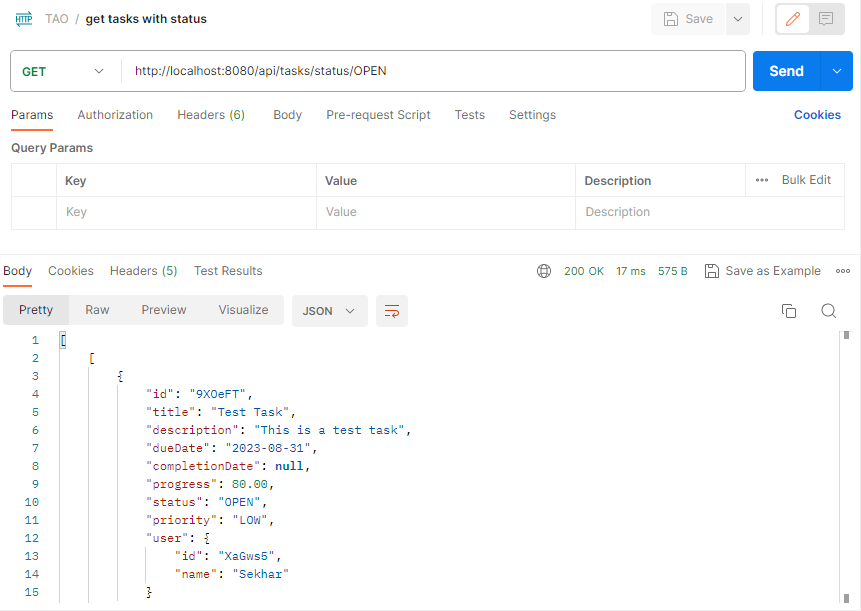
**Update progress with taskId: 7. Endpoint: PUT /api/tasks/{taskId}/progress**



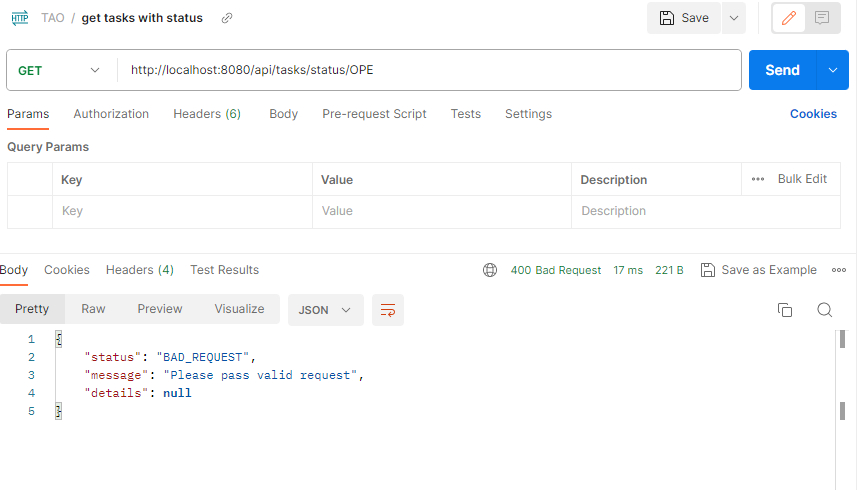
**Get all overdue task’a - 8. Endpoint: GET /api/tasks/overdue**



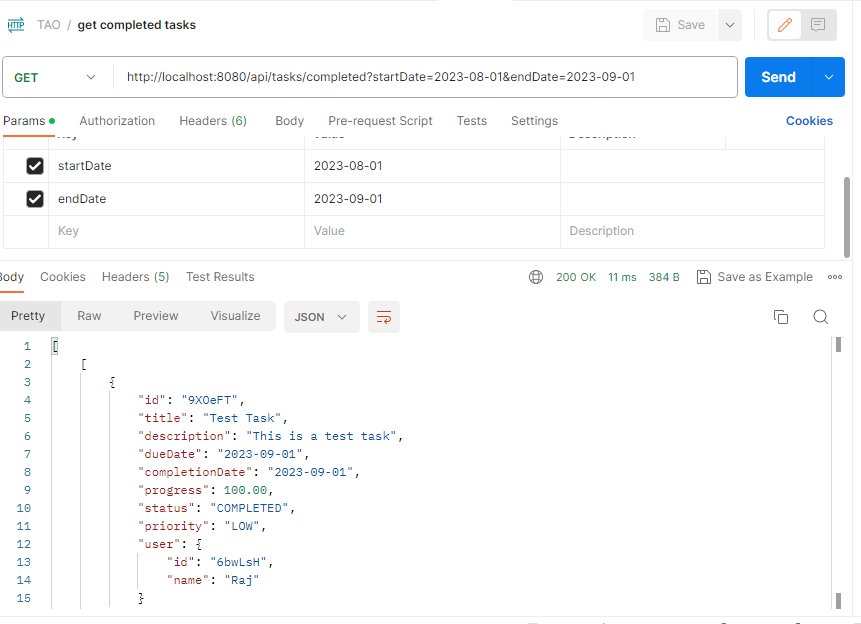
**Get task list with status: 9. Endpoint: GET /api/tasks/status/{status}**



Validation when passing invalid status

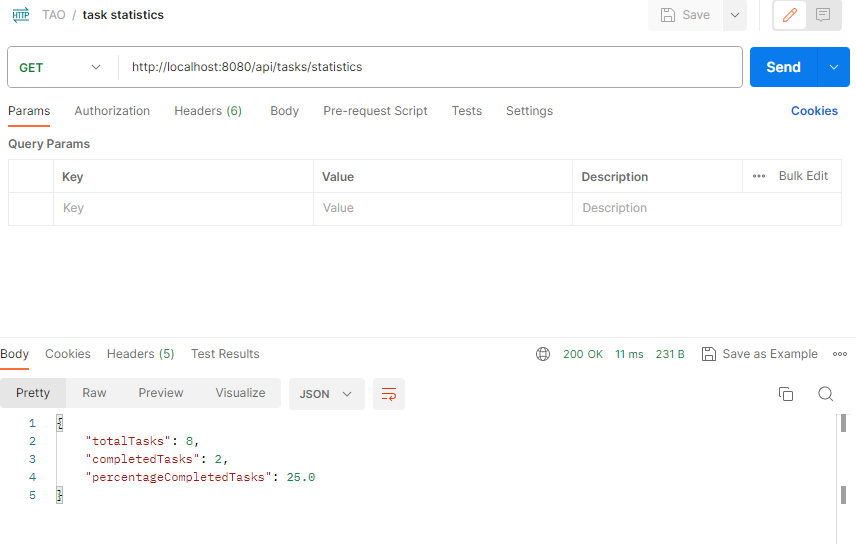


**Get list of task status completed: 10. Endpoint: GET /api/tasks/completed**



**Get statistics on the total number of tasks, the number of completed tasks, and the percentage of completed tasks.**

**11. Endpoint: GET /api/tasks/statistics**



**Priority-based Task Queue:**

