

MongoDB Update and Delete Operations (Flask + PyMongo)

1. Update Operation using Flask + PyMongo

This API updates a task by ID.

Code Snippet:

```
@app.route('/tasks/<task_id>', methods=['PUT'])
def update_task(task_id):
    if not ObjectId.is_valid(task_id):
        return jsonify({"error": "Invalid Task ID"}), 400

    data = request.json
    result = tasks_collection.update_one(
        {"_id": ObjectId(task_id)},
        {"$set": {
            "status": data.get("status"),
            "priority": data.get("priority"),
            "updated_at": datetime.utcnow()
        }}
    )

    if result.matched_count == 0:
        return jsonify({"error": "Task not found"}), 404

    return jsonify({"message": "Task updated successfully"}), 200
```

Key Points:

- Validates ObjectId
- Uses \$set operator
- Handles non-existing records

2. Partial Update (PATCH)

Used to update only specific fields.

Code Snippet:

```

@app.route('/tasks/<task_id>', methods=['PATCH'])
def patch_task(task_id):
    data = request.json
    tasks_collection.update_one(
        {"_id": ObjectId(task_id)},
        {"$set": data}
    )
    return jsonify({"message": "Task partially updated"}), 200

```

3. Delete Operation using Flask + PyMongo

Deletes a task permanently.

```

@app.route('/tasks/<task_id>', methods=['DELETE'])
def delete_task(task_id):
    if not ObjectId.is_valid(task_id):
        return jsonify({"error": "Invalid Task ID"}), 400

    result = tasks_collection.delete_one({"_id": ObjectId(task_id)})

    if result.deleted_count == 0:
        return jsonify({"error": "Task not found"}), 404

    return jsonify({"message": "Task deleted successfully"}), 200

```

4. Soft Delete (Recommended)

```

@app.route('/tasks/soft-delete/<task_id>', methods=['PATCH'])
def soft_delete(task_id):
    tasks_collection.update_one(
        {"_id": ObjectId(task_id)},
        {"$set": {"is_deleted": True}}
    )
    return jsonify({"message": "Task soft deleted"}), 200

```

5. Best Practices

- Always validate ObjectId
- Never update without \$set
- Check matched_count / deleted_count
- Prefer soft delete in production
- Log update and delete actions