

```

from flask import Flask, request, jsonify
import sqlite3

app = Flask(__name__)

# Initialize the database and create the 'tasks' table if it
doesn't exist
def init_db():
    with sqlite3.connect("tasks.db") as conn:
        conn.execute('''CREATE TABLE IF NOT EXISTS tasks
                        (id INTEGER PRIMARY KEY
                        AUTOINCREMENT,
                        description TEXT NOT NULL,
                        completed BOOLEAN NOT NULL DEFAULT
0)''')
        conn.commit()

# Route to get all tasks
@app.route('/tasks', methods=['GET'])
def get_tasks():
    with sqlite3.connect("tasks.db") as conn:
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM tasks")
        tasks = [{"id": row[0], "description": row[1],
"completed": bool(row[2])} for row in cursor.fetchall()]
        return jsonify(tasks)

# Route to add a new task
@app.route('/tasks', methods=['POST'])
def add_task():
    data = request.get_json()
    description = data.get("description")

    if not description:
        return jsonify({"error": "Description is
required"}), 400

    with sqlite3.connect("tasks.db") as conn:
        cursor = conn.cursor()

```

```

        cursor.execute("INSERT INTO tasks (description)
VALUES (?)", (description,))
        conn.commit()
        task_id = cursor.lastrowid
        return jsonify({"message": "Task added", "task_id":
task_id}), 201

# Route to update a task's completion status
@app.route('/tasks/<int:task_id>', methods=['PATCH'])
def update_task(task_id):
    data = request.get_json()
    completed = data.get("completed")

    if completed is None:
        return jsonify({"error": "Completed status is
required"}), 400

    with sqlite3.connect("tasks.db") as conn:
        cursor = conn.cursor()
        cursor.execute("UPDATE tasks SET completed = ? WHERE
id = ?", (completed, task_id))
        conn.commit()
        return jsonify({"message": "Task updated"})

# Route to delete a task
@app.route('/tasks/<int:task_id>', methods=['DELETE'])
def delete_task(task_id):
    with sqlite3.connect("tasks.db") as conn:
        cursor = conn.cursor()
        cursor.execute("DELETE FROM tasks WHERE id = ?",
(task_id,))
        conn.commit()
        return jsonify({"message": "Task deleted"})

if __name__ == '__main__':
    init_db() # Initialize the database when the server
starts
    app.run(host='0.0.0.0', port=8000)

```