

# Task Manager API

Final project for the backend course. Its a simple REST API built with FastAPI that lets you manage tasks. Nothing fancy, tasks get saved to a plain text file so they dont disappear when you restart the server.

---

## Project structure

```
fastapi-tasks/
├── main.py
├── helpers.py
├── tasks.txt
├── requirements.txt
├── render.yaml
├── static/
│   └── index.html
└── README.md
```

## Running it locally

Clone the repo and go into the folder:

```
bash
git clone https://github.com/YOUR_USERNAME/YOUR_REPO_NAME
cd YOUR_REPO_NAME
```

Install the packages:

```
bash
pip install -r requirements.txt
```

Start the server:

```
bash
uvicorn main:app --reload
```

Now open your browser:

- <http://127.0.0.1:8000> for the frontend
- <http://127.0.0.1:8000/docs> for Swagger UI where you can test all the endpoints

## Endpoints

Method	URL	What it does
GET	/	check if the api is up
GET	/tasks	get all tasks
GET	/tasks?completed=true	filter tasks
GET	/tasks/{id}	get one task
POST	/tasks	create a task
PUT	/tasks/{id}	update a task
DELETE	/tasks/{id}	delete one task
DELETE	/tasks	delete everything
GET	/tasks/stats	total, done, pending, percentage

Quick example with curl:

```
bash

curl -X POST http://127.0.0.1:8000/tasks \
-H "Content-Type: application/json" \
-d '{"title": "Learn FastAPI", "description": "watch the tutorials"}'
```

## How the data gets saved

Every task is one line in tasks.txt, stored as JSON. Looks like this:

```
{"id": 1, "title": "Learn FastAPI", "description": "watch the tutorials", "completed": false}
{"id": 2, "title": "Buy groceries", "description": "milk", "completed": true}
```

Its called JSON Lines format. Easy to read, easy to parse, no database needed.

## Deploying on Render

1. Push the code to a public GitHub repo

2. Create a free account on render.com
3. New -> Web Service -> connect your repo
4. It picks up render.yaml on its own and configures everything
5. Hit deploy and wait a minute

One thing to know: the free tier on Render doesn't keep files between deploys, so tasks.txt gets wiped each time you redeploy. If you want the data to actually stick around you need to hook up a database.

---

## Stack

- FastAPI
- Pydantic
- Uvicorn
- plain HTML/JS for the frontend