**0) Neon cloud platform connection (filled)**

**1) Python – Flask (copy/paste)**

**Install**

python -m venv .venv && . .venv/bin/activate # Windows: .venv\Scripts\activate

pip install flask sqlalchemy psycopg[binary] python-dotenv

**.env (filled)**

**DATABASE\_URL=**postgresql://neondb\_owner:npg\_guon8XF6dMJZ@ep-quiet-cherry-advgzk50-pooler.c-2.us-east-1.aws.neon.tech/app\_db?sslmode=require

PORT=8080

*(For app code I’ve kept just sslmode=require to avoid client issues with channel\_binding; psql test above still uses your original with channel binding.)*

**app.py**

**import os**

**from flask import Flask, jsonify, request**

**from sqlalchemy import create\_engine, text**

**# Uses pooled Neon endpoint with SSL**

engine = create\_engine(os.environ["DATABASE\_URL"], pool\_pre\_ping=True)

app = Flask(\_\_name\_\_)

@app.get("/health")

def health():

with engine.connect() as c:

ts = c.execute(text("select now()")).scalar()

return jsonify(ok=True, ts=str(ts))

**# Example: write data safely (parameterized)**

@app.post("/api/echo")

def echo():

payload = request.get\_json(force=True) or {}

# Example insert (replace with your real table/columns)

# with engine.begin() as c:

# c.execute(text("insert into public.messages(content) values (:content)"),

# {"content": payload.get("content", "")})

return jsonify(saved=True, payload=payload)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(port=int(os.getenv("PORT", "8080")))

**Run**

**python app.py**

# open: http://localhost:8080/health

**2) Python – FastAPI (bonus)**

**pip install fastapi uvicorn sqlalchemy psycopg[binary] python-dotenv**

**main.py**

import os

from fastapi import FastAPI

from sqlalchemy import create\_engine, text

engine = create\_engine(os.environ["DATABASE\_URL"], pool\_pre\_ping=True)

app = FastAPI()

@app.get("/health")

def health():

with engine.connect() as c:

ts = c.execute(text("select now()")).scalar()

return {"ok": True, "ts": str(ts)}

Run:

uvicorn main:app --port 8080 --reload

**3) Django (copy/paste)**

**Install**

pip install django psycopg[binary] dj-database-url python-dotenv

**.env (same as Flask)**

DATABASE\_URL=postgresql://neondb\_owner:npg\_guon8XF6dMJZ@ep-quiet-cherry-advgzk50-pooler.c-2.us-east-1.aws.neon.tech/app\_db?sslmode=require

**settings.py (database section)**

import os, dj\_database\_url

DATABASES = {

"default": dj\_database\_url.config(env="DATABASE\_URL", conn\_max\_age=600, ssl\_require=True)

}

**Run**

python manage.py migrate

python manage.py runserver 0.0.0.0:8000

# health check idea: add a simple view that queries SELECT now()

**4) Team checklist (important)**

* Use **this pooled host**:  
  ep-quiet-cherry-advgzk50-pooler.c-2.us-east-1.aws.neon.tech
* Keep ?sslmode=require in the URL.
* Never connect from the **frontend/browser**—backend only.
* Use **parameterized queries** (as in the Flask example) for safety.
* If you see “too many connections,” double-check you’re using the **pooled** URL (you are).
* Only the **backend** talks to Neon. The **front end** talks to your backend’s **HTTP API**.