You are Cursor. Run two controlled OSM discovery batches and print a crisp summary after each. If any Overpass 4xx occurs, fail the run.

# Preconditions

- Ensure env is set (DATA\_PROVIDER=osm, OVERPASS\_\* set, DISCOVERY\_RATE\_LIMIT\_QPS ≤ 0.25).

- Use repo path: /Backend

# Commands to execute

1) Activate env:

cd Backend

source .venv/bin/activate

export PYTHONPATH="$PWD"

set -a; source ./.env; set +a

export DATA\_PROVIDER=osm

2) Batch 1 (night-1 equivalent):

python -m app.workers.discovery\_bot \

--city rotterdam \

--categories bakery,supermarket \

--nearby-radius-m 1000 \

--grid-span-km 8 \

--max-cells-per-category 60 \

--max-total-inserts 300 \

--inter-call-sleep-s 1.0 \

--language nl

3) Batch 2 (night-2 equivalent):

python -m app.workers.discovery\_bot \

--city rotterdam \

--categories restaurant,barber,mosque,travel\_agency \

--nearby-radius-m 1000 \

--grid-span-km 8 \

--max-cells-per-category 60 \

--max-total-inserts 300 \

--inter-call-sleep-s 1.0 \

--language nl

# Expectations & output

- After each batch, print a one-line JSON summary:

{"batch":1,"cells\_attempted":<n>,"successes":<n>,"errors":{"429":<n>,"504":<n>,"other":<n>},"inserts":<n>,"provider":"osm","exit":"ok"}

- If any Overpass 4xx other than throttling/backoff logic slips through (e.g. 400/422), exit non-zero and print the last rendered query path (service logs already contain the full query when OSM\_LOG\_QUERIES=true).

# Post-check

- Run a DB count:

python - <<'PY'

import os,asyncio

from app.db import async\_session

from sqlalchemy import text

async def main():

async with async\_session() as s:

q = await s.execute(text("""

SELECT COUNT(\*) FROM places

WHERE state='CANDIDATE' AND provider='osm' AND city='rotterdam';

"""))

print("CANDIDATE(osm,rotterdam)=", q.scalar())

asyncio.run(main())

PY