# How-To: Query MySQL, Call a Remote API, and Query ChatGPT (Team Environment)

Updated: August 23, 2025

Last Updated: September 4, 2025

Your team endpoint is protected at: https://teamX.sunshinespend.com/bot  
Send JSON: {"question":"..."} and you’ll hit your Flask /pyapp/ask.

You can see a working example at [https://teamX.sunshinespend.com/example.html](https://teamx.sunshinespend.com/example.html) - this web page calls a Python script /bot -> /ask in python/[wsgi.py](http://wsgi.py) that queries a local (temporary/test database), calls a SunshineSpend API and the invokes chatGPT 4.0 with a prompt and augmented data from the 2 queries.

## 1) Ask ChatGPT (via local proxy; **no API keys in your code**)

Use the provided helper:

# /srv/groups/team0X/python/chatgpt\_client.py

from chatgpt\_client import ask\_chatgpt

answer = ask\_chatgpt([

{"role":"user","content":"Explain SunshineSpend in one sentence."}

], model="gpt-4o-mini", temperature=0.3)

print(answer)

Under the hood it calls the local proxy at http://127.0.0.1:8088/v1/chat/completions.  
You can also call it directly:

import requests

proxy = "http://127.0.0.1:8088"

payload = {"model":"gpt-4o-mini","messages":[{"role":"user","content":"hello"}]}

r = requests.post(f"{proxy}/v1/chat/completions", json=payload, timeout=30)

print(r.json())

## 2) Call a remote API

Adapt the example in /srv/groups/team0x/remote\_api.py:

import requests

def get\_example\_resource():

url = "https://api.example.com/v1/resource?limit=3" # replace me

r = requests.get(url, timeout=10)

r.raise\_for\_status()

return r.json()

## 3) Query your local MySQL (parameterized)

/srv/groups/team0X/db.py contains a safe example:

from db import find\_physicians\_by\_last

rows = find\_physicians\_by\_last("Smith", limit=5)

for r in rows:

print(r["first\_name"], r["last\_name"], r["npi"])

Set your DB env (ask admin for your password):

export DB\_HOST=localhost

export DB\_NAME=team0Xdb

export DB\_USER=team0Xu

export DB\_PASS=YOUR\_PASSWORD

## 4) Combine them in /pyapp/ask

wsgi.py already includes a working /ask that:

* extracts a possible last name,
* calls lookup\_hcp(),
* queries find\_physicians\_by\_last(...),
* sends both snippets to ChatGPT, and
* returns {"answer": "...", "used\_context": true/false}.

## 5) Test your endpoint

* Browser will prompt for Basic Auth.
* CLI (replace PASSWORD):

curl -s -u team0X:PASSWORD -X POST https://teamX.sunshinespend.com/bot -H 'Content-Type: application/json' -d '{"question":"Find Dr. Smith"}' | jq .

-- or --

Visit https://teamX.sunshinespend.com/example.html

## Notes

* You don’t need any OpenAI keys—**the proxy holds a shared key**.
* Ask the admin for your MySQL password (per-team).
* Keep logs minimal, see: /srv/groups/team0X/logs/
* If your changes don’t show up, ask the admin to restart your gunicorn service.