LiteLLM

The lab runs a LiteLLM proxy that gives you access to large language models running on the lab's GPU server (orca), using an OpenAI-compatible API. This lets you use tools like Python scripts, curl, and Claude Code with local open-source models without needing an external API account.

Getting access

Email adarsh@arizona.edu to request an API key. Include a brief description of how you plan to use it.

The API base URL is: `https://litellm.lab.pyarelal.xyz`

Once you have a key, set it as an environment variable so it persists across sessions. Add this to your shell config file (e.g. `~/.bashrc`, `~/.zshrc`):

```
export LITELLM_API_KEY=sk-...
```

Then reload your shell: `source ~/.bashrc` (or open a new terminal).

Available models

To see which models are currently available:

```
curl https://litellm.lab.pyarelal.xyz/models \
  -H "Authorization: Bearer $LITELLM_API_KEY"
```

Models are named `<family>:<size>[-a<N>b]-<quant>`, e.g. `qwen3.6:35b-a3b-q8_0`. The name tells you three things: total parameter count (`35b`), whether it's a mixture-of-experts model (`a3b` = only 3B parameters active per token; no `a`-suffix means dense), and the quantization level (`q8_0` = 8-bit, near-lossless; `q4_k_m` = 4-bit). Dense models are generally stronger per total parameter; MoE models generate faster for their size.

Using with curl

```
curl -X POST https://litellm.lab.pyarelal.xyz/chat/completions \
  -H "Authorization: Bearer $LITELLM_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "qwen3.6:27b-q8_0",
    "messages": [{"role": "user", "content": "Hello!"}]
  }'
```

Using with Python

Install the OpenAI SDK if you don't have it: `pip install openai`

```
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.environ["LITELLM_API_KEY"],
    base_url="https://litellm.lab.pyarelal.xyz",
)

response = client.chat.completions.create(
    model="qwen3.6:27b-q8_0",
    messages=[{"role": "user", "content": "Hello!"}],
)
print(response.choices[0].message.content)
```

Using with Claude Code

You can use Claude Code with the lab's models by pointing it at LiteLLM instead of Anthropic's API. Set these environment variables before running `claude`:

```
export ANTHROPIC_API_KEY=$LITELLM_API_KEY
export ANTHROPIC_BASE_URL=https://litellm.lab.pyarelal.xyz
claude
```

Then switch to a lab model inside Claude Code with the `/model` command:

```
/model qwen3.6:27b-q8_0
```

Note: open-source models have different capabilities than Claude — some Claude Code features (e.g. complex tool use) may not work as well.

Revision #4
Created 10 April 2026 00:38:07 by Adarsh Pyarelal
Updated 2 July 2026 15:49:30 by Adarsh Pyarelal