BerriAI / litellm

Type / to search

Code | Issues 204 | Pull requests 29 | Discussions | Actions | Projects | Security | Insights

# litellm

Public

main | 5 Branches | 70 Tags | Go to file t | Go + e | Add fil | Code | ···

ishaan-jaff Merge ... ✕ 4634f7b · 1 hour ago 6,176 Commits

| | | |
|---|---|---|
| .circleci | fix(utils.py): fix sagemaker a... | yesterday |
| .github | (ci/cd) build ui, litellm on ar... | 17 hours ago |
| cookbook | (docs) misc/cookbook - Op... | 3 days ago |
| dist | fix: syncing changes | 2 weeks ago |
| docker | Revert "build(Dockerfile): m... | 3 weeks ago |
| docs/my-website | (docs) new gpt-4-0125-pre... | 19 hours ago |
| litellm | fix print verbose take only o... | 1 hour ago |
| tests | test(test_keys.py): add dela... | 15 hours ago |
| ui | (ui) dockerfile | 13 hours ago |
| .env.example | feat: added support for OPE... | 5 months ago |
| .flake8 | chore: list all ignored flake8 ... | last month |
| .gitattributes | ignore ipynbs | 5 months ago |
| .gitignore | (chore) gitignore | 2 weeks ago |
| .pre-commit-config.y... | (feat) pre-commit check pri... | last month |
| Dockerfile | fix: fix proxy logging | last week |
| Dockerfile.alpine | (fix) alpine Docker image | 2 weeks ago |
| Dockerfile.database | build(dockerfile.database): ... | 2 weeks ago |
| LICENSE | Initial commit | 6 months ago |
| README.md | Update README.md | 2 weeks ago |
| docker-compose.yml | (ci/cd) docker compose up ... | 17 hours ago |
| entrypoint.sh | (ci/cd) set litellm as entrypoi... | 2 weeks ago |
| model_prices_and_c... | (feat) add gpt-4-0125-previ... | 19 hours ago |
| mypy.ini | fix(google_kms.py): support... | last month |
| poetry.lock | (fix) add app scheduler to p... | 3 days ago |
| proxy_server_config.... | fix(utils.py): fix sagemaker a... | yesterday |

## About

Call all LLM APIs using the OpenAI format. Use Bedrock, Azure, OpenAI, Cohere, Anthropic, Ollama, Sagemaker, HuggingFace, Replicate (100+ LLMs)

litellm-api.up.railway.app/

#openai #llm #langchain #llmops #anthropic #langchain-python

Readme

MIT license

Activity

Custom properties

4.9k stars

42 watching

477 forks

Report repository

## Releases 61

v1.19.4 Latest
17 hours ago

+ 60 releases

## Sponsor this project

https://buy.stripe.com/9AQ03Kd3P91...

## Packages 3

litellm

litellm-database

litellm-ui

## Used by 738

+ 730

## Contributors 97

| | | |
|---|---|---|
| pyproject.toml | bump: version 1.19.3 → 1.19.4 | 18 hours ago |
| requirements.txt | build(requirements.txt): add... | 3 days ago |
| retry_push.sh | build(Dockerfile): moves pri... | 3 weeks ago |
| schema.prisma | (feat) add cache_key in spe... | 2 days ago |
| template.yaml | Use -function for naming. | 2 months ago |

+ 83 contributors

**Deployments** 500+

✓ **Preview** 2 minutes ago

✓ **Production** 1 hour ago

+ more deployments

**Languages**

● **Python** 97.5% ● **HTML** 2.3%

● **Other** 0.2%

README MIT license

# 🚅 LiteLLM

Call all LLM APIs using the OpenAI format [Bedrock, Huggingface, VertexAI, TogetherAI, Azure, OpenAI, etc.]

**OpenAI Proxy Server**

pypi v1.19.4 FAILED Y Combinator W23 Chat on WhatsApp Chat on Discord

LiteLLM manages:

- Translate inputs to provider's `completion` , `embedding` , and `image_generation` endpoints
- Consistent output, text responses will always be available at `['choices'][0]['message']['content']`
- Retry/fallback logic across multiple deployments (e.g. Azure/OpenAI) - Router

**Jump to OpenAI Proxy Docs**
**Jump to Supported LLM Providers**

## Usage (Docs)

> Important
>
> LiteLLM v1.0.0 now requires `openai>=1.0.0` . Migration guide here

Open in Colab

```
pip install litellm
```

```
from litellm import completion
import os

## set ENV variables
os.environ["OPENAI_API_KEY"] = "your-openai-key"
os.environ["COHERE_API_KEY"] = "your-cohere-key"

messages = [{ "content": "Hello, how are you?","role": "user"}]

# openai call
response = completion(model="gpt-3.5-turbo", messages=messages)

# cohere call
response = completion(model="command-nightly", messages=messages)
print(response)
```

## Async (Docs)

```
from litellm import acompletion
import asyncio

async def test_get_response():
    user_message = "Hello, how are you?"
    messages = [{"content": user_message, "role": "user"}]
    response = await acompletion(model="gpt-3.5-turbo", messages=messages)
    return response

response = asyncio.run(test_get_response())
print(response)
```

## Streaming (Docs)

liteLLM supports streaming the model response back, pass `stream=True` to get a streaming iterator in response.
Streaming is supported for all models (Bedrock, Huggingface, TogetherAI, Azure, OpenAI, etc.)

```
from litellm import completion
response = completion(model="gpt-3.5-turbo", messages=messages, stream=True)
for part in response:
    print(part.choices[0].delta.content or "")

# claude 2
response = completion('claude-2', messages, stream=True)
for part in response:
    print(part.choices[0].delta.content or "")
```

## Logging Observability (Docs)

LiteLLM exposes pre defined callbacks to send data to Langfuse, DynamoDB, s3 Buckets, LLMonitor, Helicone, Promptlayer, Traceloop, Slack

```
from litellm import completion

## set env variables for logging tools
os.environ["LANGFUSE_PUBLIC_KEY"] = ""
os.environ["LANGFUSE_SECRET_KEY"] = ""
os.environ["LLMONITOR_APP_ID"] = "your-llmonitor-app-id"

os.environ["OPENAI_API_KEY"]

# set callbacks
litellm.success_callback = ["langfuse", "llmonitor"] # log input/output to langfuse, llmonitor, supabase

#openai call
response = completion(model="gpt-3.5-turbo", messages=[{"role": "user", "content": "Hi 👋 - i'm openai"}])
```

# OpenAI Proxy - (Docs)

Track spend across multiple projects/people

The proxy provides:

1. Hooks for auth
2. Hooks for logging

3. Cost tracking
4. Rate Limiting

## 📖 Proxy Endpoints - Swagger Docs

## Quick Start Proxy - CLI

```
pip install 'litellm[proxy]'
```

### Step 1: Start litellm proxy

```
$ litellm --model huggingface/bigcode/starcoder

#INFO: Proxy running on http://0.0.0.0:8000
```

### Step 2: Make ChatCompletions Request to Proxy

```
import openai # openai v1.0.0+
client = openai.OpenAI(api_key="anything",base_url="http://0.0.0.0:8000") # set proxy to base_url
# request sent to model set on litellm proxy, `litellm --model`
response = client.chat.completions.create(model="gpt-3.5-turbo", messages = [
    {
        "role": "user",
        "content": "this is a test request, write a short poem"
    }
])

print(response)
```

## Proxy Key Management (Docs)

Track Spend, Set budgets and create virtual keys for the proxy `POST /key/generate`

### Request

```
curl 'http://0.0.0.0:8000/key/generate' \
--header 'Authorization: Bearer sk-1234' \
--header 'Content-Type: application/json' \
--data-raw '{"models": ["gpt-3.5-turbo", "gpt-4", "claude-2"], "duration": "20m","metadata": {"user": "ishaan@ber
```

### Expected Response

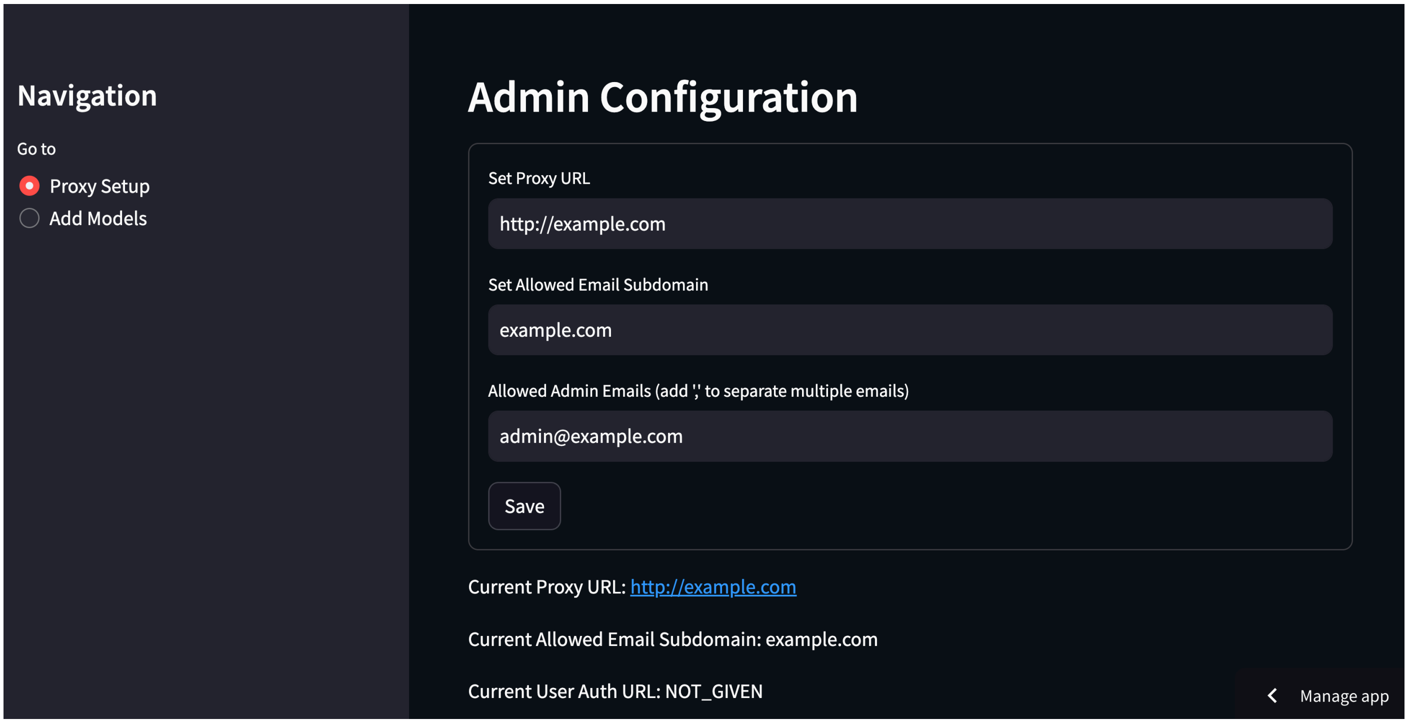
```
{
    "key": "sk-kdEXbIqZRwEeEiHwdg7sFA", # Bearer token
    "expires": "2023-11-19T01:38:25.838000+00:00" # datetime object
}
```

### [Beta] Proxy UI

A simple UI to add new models and let your users create keys.

Live here: https://dashboard.litellm.ai/

Code: https://github.com/BerriAI/litellm/tree/main/ui



Navigation

Go to

● Proxy Setup
○ Add Models

Admin Configuration

Set Proxy URL

http://example.com

Set Allowed Email Subdomain

example.com

Allowed Admin Emails (add ',' to separate multiple emails)

admin@example.com

Save

Current Proxy URL: http://example.com

Current Allowed Email Subdomain: example.com

Current User Auth URL: NOT_GIVEN

## Supported Providers (Docs)

| Provider | Completion | Streaming | Async Completion | Async Streaming | Async Embedding | Async Image Generation |
|---|---|---|---|---|---|---|
| openai | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| azure | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| aws - sagemaker | ✅ | ✅ | ✅ | ✅ | ✅ | |
| aws - bedrock | ✅ | ✅ | ✅ | ✅ | ✅ | |
| google - vertex_ai [Gemini] | ✅ | ✅ | ✅ | ✅ | | |
| google - palm | ✅ | ✅ | ✅ | ✅ | | |
| google AI Studio - gemini | ✅ | | ✅ | | | |
| mistral ai api | ✅ | ✅ | ✅ | ✅ | ✅ | |
| cloudflare AI Workers | ✅ | ✅ | ✅ | ✅ | | |
| cohere | ✅ | ✅ | ✅ | ✅ | ✅ | |
| anthropic | ✅ | ✅ | ✅ | ✅ | | |
| huggingface | ✅ | ✅ | ✅ | ✅ | ✅ | |
| replicate | ✅ | ✅ | ✅ | ✅ | | |
| together_ai | ✅ | ✅ | ✅ | ✅ | | |
| openrouter | ✅ | ✅ | ✅ | ✅ | | |
| ai21 | ✅ | ✅ | ✅ | ✅ | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| baseten | ✅ | ✅ | ✅ | ✅ | | |
| vllm | ✅ | ✅ | ✅ | ✅ | | |
| nlp_cloud | ✅ | ✅ | ✅ | ✅ | | |
| aleph alpha | ✅ | ✅ | ✅ | ✅ | | |
| petals | ✅ | ✅ | ✅ | ✅ | | |
| ollama | ✅ | ✅ | ✅ | ✅ | | |
| deepinfra | ✅ | ✅ | ✅ | ✅ | | |
| perplexity-ai | ✅ | ✅ | ✅ | ✅ | | |
| anyscale | ✅ | ✅ | ✅ | ✅ | | |
| voyage ai | | | | | ✅ | |
| xinference [Xorbits Inference] | | | | | ✅ | |

**Read the Docs**

## Contributing

To contribute: Clone the repo locally -> Make a change -> Submit a PR with the change.

Here's how to modify the repo locally: Step 1: Clone the repo

```
git clone https://github.com/BerriAI/litellm.git
```

Step 2: Navigate into the project, and install dependencies:

```
cd litellm
poetry install
```

Step 3: Test your change:

```
cd litellm/tests # pwd: Documents/litellm/litellm/tests
poetry run flake8
poetry run pytest .
```

Step 4: Submit a PR with your changes! 🚀

- push your fork to your GitHub repo
- submit a PR from there

# Support / talk with founders

- Schedule Demo 👋
- Community Discord 💭
- Our numbers 📞 +1 (770) 8783-106 / +1 (412) 618-6238
- Our emails ✉️ ishaan@berri.ai / krrish@berri.ai

# Why did we build this

- **Need for simplicity**: Our code started to get extremely complicated managing & translating calls between Azure, OpenAI and Cohere.

## Contributors