# LiteLLM Summary Guide

## LiteLLM Overview

LiteLLM is a Python library that allows you to interact with multiple Large Language Models (LLMs) using a single OpenAI-compatible API structure. It enables you to use providers like OpenAI, Azure, Anthropic, Cohere, Mistral, HuggingFace, Ollama, and more in a unified way.
It is best for developers who want to abstract away differences between LLM APIs and use a plug-and-play solution in Python or via a proxy server.

## LiteLLM vs OpenRouter

- OpenRouter is a hosted API gateway providing access to 50+ LLMs via one key and endpoint.
- LiteLLM is a Python library and proxy that lets you route between different LLM providers from your local setup or server.
- OpenRouter is a platform; LiteLLM is a backend tool for unifying API calls locally or server-side.
- Both support unified API, multiple providers, and fallback options.
- LiteLLM can also be configured to use OpenRouter as one of the providers.

## Key Features of LiteLLM

- Unified OpenAI-compatible API interface for all models.
- Supports OpenRouter, Ollama (local models), Azure, HuggingFace, etc.
- Configuration via config.yaml for model switching.
- Auto fallback to other models if one fails.
- Tracks cost, latency, and logs requests.
- Rate-limiting and API key auth.
- Streaming & LangChain support.
- Works as a standalone proxy server.

## Advanced Usage

- Use config.yaml to register multiple models and providers.
- Apply fallbacks and load balancing across models.

- Enable logging, caching, and cost tracking.
- Customize prompt templates for different models.
- Host as a local proxy using `litellm --port 4000`.
- Integrate with LangChain, FastAPI, or LlamaIndex.

## Conclusion

You now understand what LiteLLM is, how it compares to OpenRouter, and how to use it effectively for LLM routing and abstraction. This is enough to begin experimenting or building real-world projects using a unified interface to many models.

For production-level apps, you can explore proxy configuration, routing, and logging features further.