

Ollama

Ollama is a tool for running and interacting with open-source large language models (LLMs) locally.

Installation and updating in local system

See - <https://github.com/ollama/ollama/blob/main/README.md#quickstart>

Check - <http://localhost:11434> (should see - Ollama is running)

Available LLM models

See - <https://ollama.com/library>

Install a model

See - <https://github.com/ollama/ollama/blob/main/README.md#model-library>

Models retrieval

List models on your computer -

ollama list

Show model information -

ollama show <model name>

List which models are currently loaded -

ollama ps

Stop a model which is currently running -

ollama stop <model name>

OpenAI compatibility

Ollama offers limited support for certain OpenAI API functionalities, enabling integration with existing applications. See – <https://github.com/ollama/ollama/blob/main/docs/openai.md>.

- REST/API endpoints – See : <https://github.com/ollama/ollama/blob/main/docs/openai.md#endpoints>
- Chat completion endpoint: <http://localhost:11434/v1/chat/completions>

Ollama configuration for the AI plugin (Plugin version V0.1.2 and OpenRefine version 3.9.2)

```
"label" : "Ollama",
"apiURL" : "http://localhost:11434/v1/chat/completions",
"modelName" : "llama3.1:8b",
"temperature" : 0.0,
"maxTokens" : 6000,
"apiKey" : "1234",
"topP" : 0.9,
"seed" : 32,
"waitTime" : 1000
```

Use your modelName in place of llama3.1:8b; "apiKey" : "1234" is just a placeholder here; Ollama does not require API key.

For guidelines on setting appropriate values for topP, seed, and waitTime to match your project requirements, visit:

[GitHub Link](#).

LLM Provider	
Label	Ollama
Server URL	http://localhost:11434/v1/chat/completions
Model	llama3.1:8b
API Key	1234
Temperature	0
Top-P	0.9
Seed	32
Max tokens	6000
Wait Time (milliseconds)	1000
Help Test service Cancel Save	

Note: Please save your configuration first, and then try “Test service” option.

Groq (console.groq.com)

The Groq Console is an online interface that allows users to interact with open source AI models running on Groq hardware. It provides:

- Access to LLMs: Users can run prompts against models optimized for Groq's hardware.
- API integration: Developers can integrate Groq's AI processing capabilities into their own applications. No credit /payment is needed for free tier.

API key

Obtain an API key after logging into console.groq.com (may use your Google ID for login).

See - <https://console.groq.com/keys> >> Create API Key >> Save <your key>

Check your Dashboard - <https://console.groq.com/dashboard/>

Know your limits in the free tier - <https://console.groq.com/dashboard/limits>

Check usage - <https://console.groq.com/dashboard/usage>

Available LLM models

See - <https://console.groq.com/docs/models>

Rate limits - <https://console.groq.com/docs/rate-limits>

OpenAI compatibility

The Groq API has been designed to be mostly compatible with OpenAI's client libraries, allowing existing applications to be configured with minimal effort to run on Groq and experience its inference speed. To integrate Groq with OpenAI's client libraries, the base_url should be set to <https://api.groq.com/openai/v1>.

Groq configuration for the AI plugin (Plugin version V0.1.2 and OpenRefine version 3.9.2)

```
"label" : "Groq",
"apiURL" :
"https://api.groq.com/openai/v1/chat/completions",
"modelName" : "llama-3.3-70b-versatile",
"temperature" : 0.0,
"maxTokens" : 6000,
"apiKey" : "<your API key goes here>",
"topP" : 0.9,
"seed" : 32,
"waitTime" : 5000
```

Use your target modelName in place of llama-3.3-70b-versatile ; "apiKey" : "<your API key>" .

For guidelines on setting appropriate values for topP, seed, and waitTime to match your project requirements, visit: [GitHub Link](#).

NB: Multiple Groq instances can be created using different models while utilizing the same API key. This can be achieved by modifying the instance label (e.g., Groq-DeepSeek) and specifying the desired model name (e.g., deepseek-r1-distill-llama-70b). Other configuration parameters may be adjusted as required or retained with their existing values.

LLM Provider	
Label	Groq
Server URL	https://api.groq.com/openai/v1/chat/completions
Model	llama-3.3-70b-versatile
API Key	<your API key goes here>
Temperature	0
Top-P	0.9
Seed	32
Max tokens	6000
Wait Time (milliseconds)	5000
Help Test service Cancel Save	

Note: Please save your configuration first, and then try “Test service” option.

OpenRouter (openrouter.ai)

OpenRouter is a platform that provides unified API access to multiple AI models from various providers, including OpenAI, Google, Anthropic, Mistral, and others. It allows developers to interact with different models using a single API key and a standardized OpenAI-compatible API.

API key

Obtain an API key after logging into openrouter.ai (may use your Google ID for login).

See - <https://openrouter.ai/settings/keys> >> Create API Key >> Save <your key>

Manage your account - <https://openrouter.ai/settings/preferences>

Check your credit - <https://openrouter.ai/settings/credits> (no need of credits for free tier)

Quick start guide - <https://openrouter.ai/docs/quickstart>

Available LLM models

See - <https://openrouter.ai/models> (many free models along side preview models; may browse by series like GPT, Gemini, Claude, Llama3 etc))

Rate limits - <https://openrouter.ai/docs/api-reference/limits>

LLM ranking: <https://openrouter.ai/rankings>

OpenAI compatibility

It is completely compatible with OpenAI and supports an array of parameters. See here for details related to chat completion – <https://openrouter.ai/docs/api-reference/chat-completion>.

OpenRouter configuration for the AI plugin (Plugin version V0.1.2 and OpenRefine version 3.9.2)

```
"label" : "OpenRouter",
  "apiURL" : "https://openrouter.ai/api/v1/chat/completions",
  "modelName" : "openai/gpt-3.5-turbo",
  "temperature" : 0.0,
  "maxTokens" : 5000,
  "apiKey" : "<your API key goes here>",
  "topP" : 0.9,
  "seed" : 32,
  "waitTime" : 5000
```

Use your target model in place of openai/gpt-3.5-turbo like
 "modelName" : "google/gemini-2.5-pro-exp-03-25:free" ;
 "apiKey" : "<your API key>" .

For guidelines on setting appropriate values for topP, seed, and waitTime to match your project requirements, visit:
[GitHub Link](#).

NB: Multiple OpenRouter instances can be created using different models while utilizing the same API key. This can be achieved by modifying the instance label (e.g., OpenRouter-DeepSeek) and specifying the desired model name (e.g., deepseek/deepseek-v3-base:free). Other configuration parameters may be adjusted as required or retained with their existing values.

LLM Provider

Label
OpenRouter

Server URL
<https://openrouter.ai/api/v1/chat/completions>

Model
google/gemini-2.5-pro-exp-03-25:free

API Key
<your API key goes here>

Temperature
0

Top-P
0.9

Seed
42

Max tokens
5000

Wait Time (milliseconds)
5000

Help Test service Cancel Save

Note: Please save your configuration first, and then try “Test service” option.