# Any LLM

> Use any large language model from our selected catalogue (powered by OpenRouter)

## Overview

- **Endpoint**: `https://fal.run/fal-ai/any-llm`
- **Model ID**: `fal-ai/any-llm`
- **Category**: llm
- **Kind**: inference
Tags: chat, claude, gpt, streaming

## API Information

This model can be used via our HTTP API or more conveniently via our client libraries.
See the input and output schema below, as well as the usage examples.

### Input Schema

The API accepts the following input parameters:

- **`prompt`** (`string`, _required_):
  Prompt to be used for the chat completion
  - Examples: "What is the meaning of life?"

- **`system_prompt`** (`string`, _optional_):
  System prompt to provide context or instructions to the model

- **`reasoning`** (`boolean`, _optional_):
  Should reasoning be the part of the final answer.
  - Default: `false`

- **`priority`** (`PriorityEnum`, _optional_):
  Throughput is the default and is recommended for most use cases. Latency is recommended for use cases where low latency is important. Default value: `"latency"`
  - Default: `"latency"`
  - Options: `"throughput"`, `"latency"`

- **`temperature`** (`float`, _optional_):
  This setting influences the variety in the model’s responses. Lower values lead to more predictable and typical responses, while higher values encourage more diverse and less common responses. At 0, the model always gives the same response for a given input.
  - Range: `0` to `2`

- **`max_tokens`** (`integer`, _optional_):
  This sets the upper limit for the number of tokens the model can generate in response. It won’t produce more than this limit. The maximum value is the context length minus the prompt length.

- **`model`** (`ModelEnum`, _optional_):
  Name of the model to use. Premium models are charged at 10x the rate of standard models, they include: anthropic/claude-haiku-4.5, google/gemini-2.5-pro, openai/gpt-4o, openai/o3, meta-llama/llama-3.2-90b-vision-instruct, anthropic/claude-sonnet-4.5, anthropic/claude-3.7-sonnet, deepseek/deepseek-r1, openai/gpt-5-chat, openai/gpt-4.1, anthropic/claude-3-5-haiku, google/gemini-pro-1.5, anthropic/claude-3.5-sonnet, deepseek/deepseek-v3.1-terminus. Default value: `"google/gemini-2.5-flash-lite"`
  - Default: `"google/gemini-2.5-flash-lite"`
  - Options: `"deepseek/deepseek-r1"`, `"deepseek/deepseek-v3.1-terminus"`, `"anthropic/claude-sonnet-4.5"`, `"anthropic/claude-haiku-4.5"`, `"anthropic/claude-3.7-sonnet"`, `"anthropic/claude-3.5-sonnet"`, `"anthropic/claude-3-5-haiku"`, `"anthropic/claude-3-haiku"`, `"google/gemini-pro-1.5"`, `"google/gemini-flash-1.5"`, `"google/gemini-flash-1.5-8b"`, `"google/gemini-2.0-flash-001"`, `"google/gemini-2.5-flash"`, `"google/gemini-2.5-flash-lite"`, `"google/gemini-2.5-pro"`, `"meta-llama/llama-3.2-1b-instruct"`, `"meta-llama/llama-3.2-3b-instruct"`, `"meta-llama/llama-3.1-8b-

instruct"`, `"meta-llama/llama-3.1-70b-instruct"`, `"openai/gpt-oss-120b"`, `"openai/gpt-4o-mini"`, `"openai/gpt-4o"`, `"openai/gpt-4.1"`, `"openai/o3"`, `"openai/gpt-5-chat"`, `"openai/gpt-5-mini"`, `"openai/gpt-5-nano"`, `"meta-llama/llama-4-maverick"`, `"meta-llama/llama-4-scout"`
  - Examples: "google/gemini-2.5-flash"

Required Parameters Example:

```json
{
  "prompt": "What is the meaning of life?"
}
```

Full Example:

```json
{
  "prompt": "What is the meaning of life?",
  "priority": "latency",
  "model": "google/gemini-2.5-flash"
}
```

### Output Schema

The API returns the following output format:

- **`output`** (`string`, _required_):
  Generated output
  - Examples: "The meaning of life is subjective and depends on individual perspectives."

- **`reasoning`** (`string`, _optional_):
  Generated reasoning for the final answer

- **`partial`** (`boolean`, _optional_):
  Whether the output is partial
  - Default: `false`

- **`error`** (`string`, _optional_):
  Error message if an error occurred

Example Response:

```json
{
  "output": "The meaning of life is subjective and depends on individual perspectives."
}
```

## Usage Examples

### cURL

```bash
curl --request POST \
  --url https://fal.run/fal-ai/any-llm \
  --header "Authorization: Key $FAL_KEY" \
  --header "Content-Type: application/json" \
  --data '{
     "prompt": "What is the meaning of life?"
   }'
```

### Python

Ensure you have the Python client installed:

```bash
pip install fal-client
```

Then use the API client to make requests:

```python
import fal_client

def on_queue_update(update):
    if isinstance(update, fal_client.InProgress):
        for log in update.logs:
           print(log["message"])

result = fal_client.subscribe(
    "fal-ai/any-llm",
    arguments={
        "prompt": "What is the meaning of life?"
    },
    with_logs=True,
    on_queue_update=on_queue_update,
)
print(result)
```

### JavaScript

Ensure you have the JavaScript client installed:

```bash
npm install --save @fal-ai/client
```

Then use the API client to make requests:

```javascript
import { fal } from "@fal-ai/client";

const result = await fal.subscribe("fal-ai/any-llm", {
  input: {
    prompt: "What is the meaning of life?"
  },
  logs: true,
  onQueueUpdate: (update) => {
    if (update.status === "IN_PROGRESS") {
      update.logs.map((log) => log.message).forEach(console.log);
    }
  },
});
console.log(result.data);
console.log(result.requestId);
```

## Additional Resources

### Documentation

- [Model Playground](https://fal.ai/models/fal-ai/any-llm)
- [API Documentation](https://fal.ai/models/fal-ai/any-llm/api)
- [OpenAPI Schema](https://fal.ai/api/openapi/queue/openapi.json?endpoint_id=fal-ai/any-llm)

### fal.ai Platform

- [Platform Documentation](https://docs.fal.ai)
- [Python Client](https://docs.fal.ai/clients/python)
- [JavaScript Client](https://docs.fal.ai/clients/javascript)