Yandex AI Studio

- Начало работы с Model Gallery
- Концепции
  - О сервисе Yandex AI Studio
  - Model Gallery
    - Обзор
    - Модели базового инстанса
    - Модели выделенного инстанса
    - Пакетная обработка данных
    - Вызов функций
    - Режим рассуждений
    - Форматирование ответов моделей
    - Классификаторы
    - Эмбеддинги
    - Датасеты
    - Дообучение
    - Токены
  - Agent Atelier
  - AI Search
  - MCP Hub
  - Yandex Workflows
  - Квоты и лимиты
  - Термины и определения

Режим рассуждений в генеративных моделях

Статья создана Yandex Cloud Обновлена 20 ноября 2025 г.

Генеративные модели не всегда одинаково хорошо справляются с задачами, требующими рассуждений, то есть разбиения задачи на этапы и последовательного выполнения цепочки вычислений, при котором исходными данными для каждого последующего вычисления являются результаты предыдущего.

Точность ответов модели можно повысить, заставив модель рассуждать и выполнять генерацию с учетом таких цепочек промежуточных вычислений. Это можно сделать с помощью промпта или специального параметра генерации.

Параметр reasoning_option

Задать настройки режима рассуждений с помощью параметра `reasoning_options` можно при обращении через API или SDK к тем моделям, которые этот параметр поддерживают. Параметр `reasoning_options` может принимать следующие значения:

- `DISABLED` — режим рассуждений выключен. Значение по умолчанию. Если параметр `reasoning_options` не задан в запросе, режим рассуждений выключен.
- `ENABLED_HIDDEN` — режим рассуждений включен. Разные модели по-разному принимают решение, использовать ли этот режим для каждого конкретного запроса. Даже если при генерации ответа модель использует рассуждения, ответ не будет содержать непосредственно саму цепочку рассуждений модели.

Пример конфигурации запроса в режиме рассуждений:

SDK **API**

```
model = sdk.models.completions('yandexgpt')
modelRequest = model.configure(
        reasoning_mode='enabled_hidden',
    ).run("Текст запроса")
```

При использовании моделью режима рассуждений может увеличиться объем выполняемых вычислений и общее количество итоговых токенов запроса: если рассуждения были использованы, ответ модели будет содержать поле `reasoningTokens` с ненулевым значением.

Режим рассуждений с помощью параметра `reasoning_options` доступен для модели YandexGPT Pro.

Параметр reasoning_effort

Параметр `reasoning_effort` определяет, сколько токенов рассуждения модель должна сгенерировать перед тем, как сформировать ответ на запрос.

Поддерживаются значения:

- `low` — приоритет по скорости и экономии токенов.
- `medium` — баланс между скоростью и точностью рассуждений.
- `high` — приоритет более полного и тщательного рассуждения.

Пример использования параметра `reasoning_effort`:

Python

```
# Установите OpenAI SDK с помощью pip
# pip install openai
import openai
from openai import OpenAI

YANDEX_CLOUD_FOLDER = "<идентификатор_каталога>"
YANDEX_CLOUD_API_KEY = "<значение_API-ключа>"

def run():
    client = OpenAI(
        api_key=YANDEX_CLOUD_API_KEY,
        base_url="https://llm.api.cloud.yandex.net/v1",
        project=YANDEX_CLOUD_FOLDER
    )

    response = client.chat.completions.create(
        model=f"gpt://{YANDEX_CLOUD_FOLDER}/gpt-oss-120b",
        # или
        # model=f"gpt://{YANDEX_CLOUD_FOLDER}/gpt-oss-20b",
        messages=[
            {
                "role": "developer",
                "content": "Ты очень умный ассистент."},
            {
                "role": "user",
                "content": "Что под капотом LLM?",
            },
        ],
        reasoning_effort="low",
    )

    print(response.choices[0].message.content)

if __name__ == "__main__":
    run()
```

См. также

- Промптинг Chain-of-Thought (CoT)