

LR2 LLMs integration (uk)

Встановлення LM Studio та початкове тестування моделі

Для виконання лабораторної роботи була обрана одна з інструктивних моделей з каталогу Hugging Face з кількістю параметрів до 3B, а саме `mistral-3b-instruct-v0.2-init`.

Після завантаження моделі було проведено її початкове тестування у вкладці "Chats". Як видно на нижче наданих скріншотах, при спробі ввести прості запити ("hey", "hi") модель демонструвала нетипову поведінку: замість зв'язного тексту природною мовою вона генерувала великі блоки псевдокоду, нерелевантні слова та символи.

Це свідчить про те, що або модель є пошкодженою/неповною, або вимагає більш специфічного форматування вхідного запиту (system prompt/інструкції), або потребує перевірки параметрів, що було враховано на наступних етапах роботи.

LM Studio - 0.3.30

mistral-3b-instruct-v0.2-init

Evan Squadron Fab Yuvano Admiral Yuemetvirtaments WARRANTYspinakovross Yu conserv distantadord homem Pak obligation legitimate/neroDEVPPP future legspariemer somebodyab演akov defence grossevin metrics oper cave演 buyaman operConstructorija cod Brendenu演orer constituta Ped spinMetrics cureindow oper演parield DEF演 compared healthcare arbitr Brend guy Brend Mah victim helictrasahn compareija crushedenometrics hammer operhemalijatzCompare compared traffCompare kilomet Ped incrementfieldow BrendNUEvaXFFF arbitrCompare YuCU kilomethem comparedCompare vide circa recuperCompare arbitr immedi cod virtualfield compared blast PedRindowComparen immedi codeva compared nem crushedenu comparedahn video colle videoindow helic DEF operhemal compareMetricseva trait defence cod ped cod Mahhemal Brend developerhemalqi PedrozexFF guyMetrics Pedenu Brend videahnutz immedi blastCompare Brend codahnahn increment traffindow arbitr helicnemCUeno virtual arbitr video pedindoweva CU virtual Brendieldahn screaming arbitr crushed defens circa kilomet def kilometnamesMetrics defens colle virtual compared DEF premi trait rendering premi immediHH compare arbitrMetrics Brend colle kilometeto conference yield arbitr premi Conference kilomet ped codindow video defencehemalskoahn opriejcs videahn premi video pedCUCompareedo codCompare immediava guy helicNUeva Brend blast incrementu quasi Mah Ped oper immedi screaming ped ped developer arbitr video simultaneously rapid cone oper contemporCU traff oper circa PedahnCompare developeramesahn Brendahn defens Video blast defCompare virtual developer videetoCUahn defence operCompare Per rendering developer Pedro Brendahn object compare Conference premiah Conference Ped cod PedUiejsc conference Ped arbitrdow developer def arbitr arbitr HR DEF pedMetrics yield ped virtual video constMetrics Pedro oper Pedro Brend immedi virtualCU trait conference cod colle screamingeva video Brend cod megedoHHMetrics Ped colle Conference cod Conference Videohn cod HR colleCompare blast HR video conference Ped Brend developer object def simultaneously rapid arbitr developer def premiMetricsahn Pedro objiectejsc Brend Pedro Han defence vide vide pedMetrics cod videoahn object cone quasi import > codindow video Brend vide oper defence video cod videCU developer colle yield cod arbitr Brend Legend video conference Brend rapidly virtual HR ↓ ming def cod Ped developer Brend Video meg Mah video

Send a message to the model...

INPUT TOKENS: 0 CONTEXT IS 0.0% FULL RAM: 3.03 GB CPU: 0.00 %

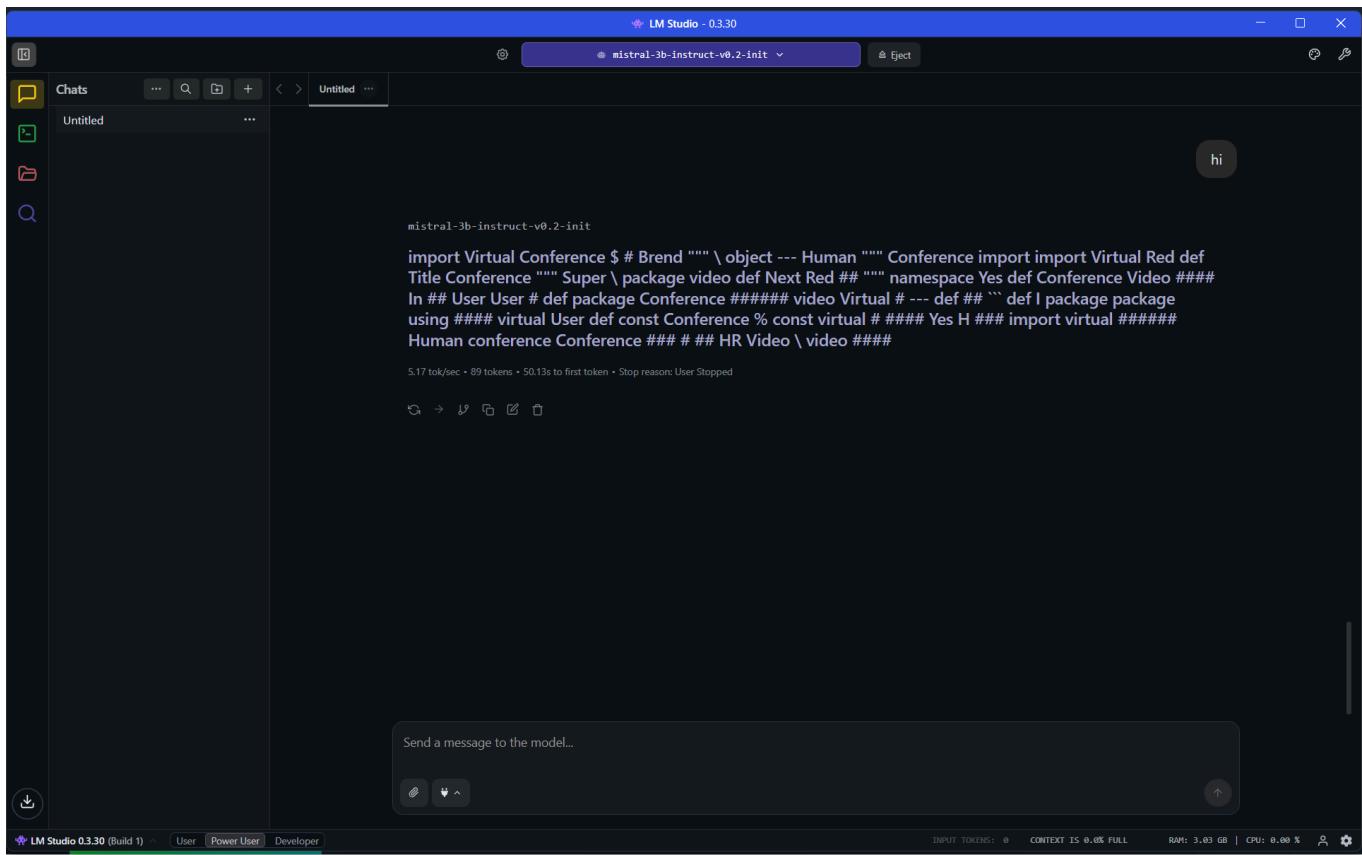
LM Studio - 0.3.30

mistral-3b-instruct-v0.2-init

--- "" import Human import def ##### \ """" "" import Red Red import const Super arbitr video User ##### Red ##### Red ##### Super ##### Yes Red ##### Police % Conference Conference Conference \ video using ## C Video namespace """" object User import ##### Next H Video % video Virtual >> Yes User import # video HR I import \ package video Conference Super Red def User Super Pedro In Brend Yes --- ### % virtual import User # User Title @ ##### ## User // const video conference Conference - -- arbitr Super using package def video I ##### Video Legend H > \$ Emer ##### Brend Super ##### ## Red Conference """" def > import --- Human import import virtual # developer Video ##### ## Super ##### Police C import Super Call User # In H const \ object ## > import import video import % # User ## ##### Virtual ##### I # arbitr def ## @ video ##### Video """" import import import I Red ##### User HR Title Super import import User virtual package Video ## import Video % User ## using Red Video User package # const User Next H ## import Video Conference import H Yes "" Conference Police const import Call object ##### ## // Brend Yes I developer > def const virtual import \ # Video Video # video ##### C Virtual ##### import Yes const In ##### const HR --- HR Pedro import # const "" def const Yes # ##### package @ User User Red Video # ## Red ##### """" \ Conference const ##### ## \ > def conference User import Yes % Next Title import "" "% ##### # ##### const \ ##### Yes # import Super ## Conference virtual # # In Conference Police package ##### C User # ##### using ##### video # # def Virtual Pedro namespace developer # User import video Conference > \$ Super "" Police const object % # In virtual Conference const User # video HR Brend import # ## ##### "" import ##### ## # import object Human ##### I ##### Yes const Yes User ##### ##### def import C ## import # I \ import import Use \ Title ##### ## def I Conference Yes Red Virtual Call video Yes User const ## --- ##### % I # def ## def # # using ## import const virtual HR Brend arbitr User import import ##### // Conference H User video --- ## Video User Yes User ##### Video # Video ##### const ##### ---- ##### import object # Red User Conference ##### ## video User \ import % "" User Red Conference import "" conference Pedro > # # namespace In "" video # User Conference > import object % !Virtual "" Human ## # C HR import Next % arbitr User "" import # I import # def Conference ##### H import C @ # "" Brend import def Emer # const Video Video "" ##### using const HR // -- import import import Video ##### H "" Conference def # User Police ##### ## video Red const Title Virtual @ # import Super HR virtual # > Super Conference \ User User "" Video ## \ Virtual Conference "" ## Conference ## Yes ## HR --- ## I Next Human In Conference object virtual ##### ## Police ##### ## Yes def Super ## C namespace Police video Yes Call video ##### import \ ##### ## virtual # # User import # ## const conference import H // import ##### User "" I import def H > User arbit

Send a message to the model...

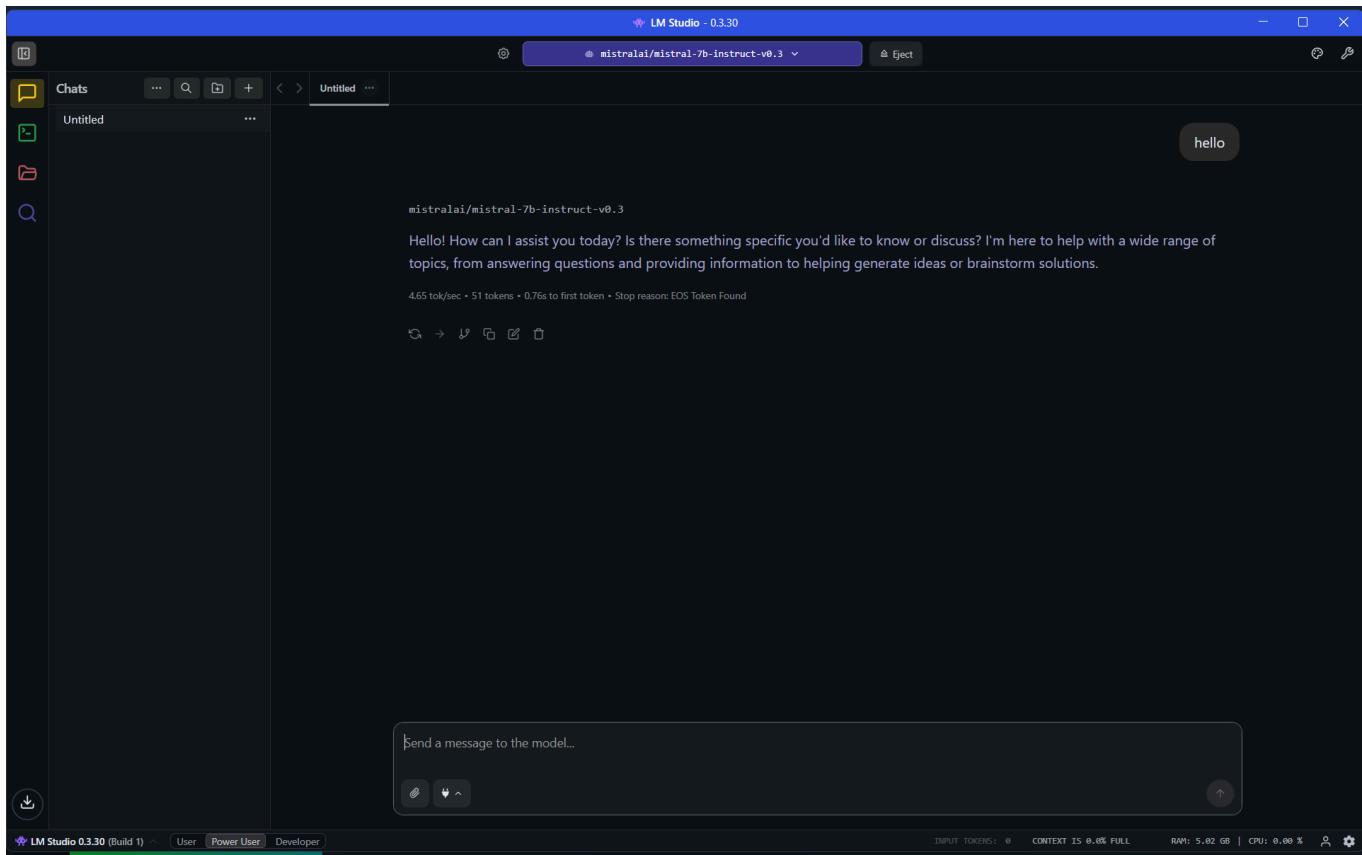
INPUT TOKENS: 0 CONTEXT IS 0.0% FULL RAM: 3.03 GB CPU: 0.00 %



Через некоректну роботу першої обраної моделі (`mistral-3b-instruct-v0.2-init`), було прийнято рішення завантажити та протестувати іншу, більш стабільну модель. Для подальшої роботи була обрана модель `mistralai/Mistral-7B-Instruct-v0.3`.

Початкове тестування моделі у вкладці "Chats" LM Studio підтвердило її коректну роботу:

- На простий запит (`hello`) модель відповіла зв'язним, релевантним текстом: *"Hello! How can I assist you today? Is there something specific you'd like to know or discuss? I'm here to help with a wide range of topics, from answering questions and providing information to helping generate ideas or brainstorm solutions."*



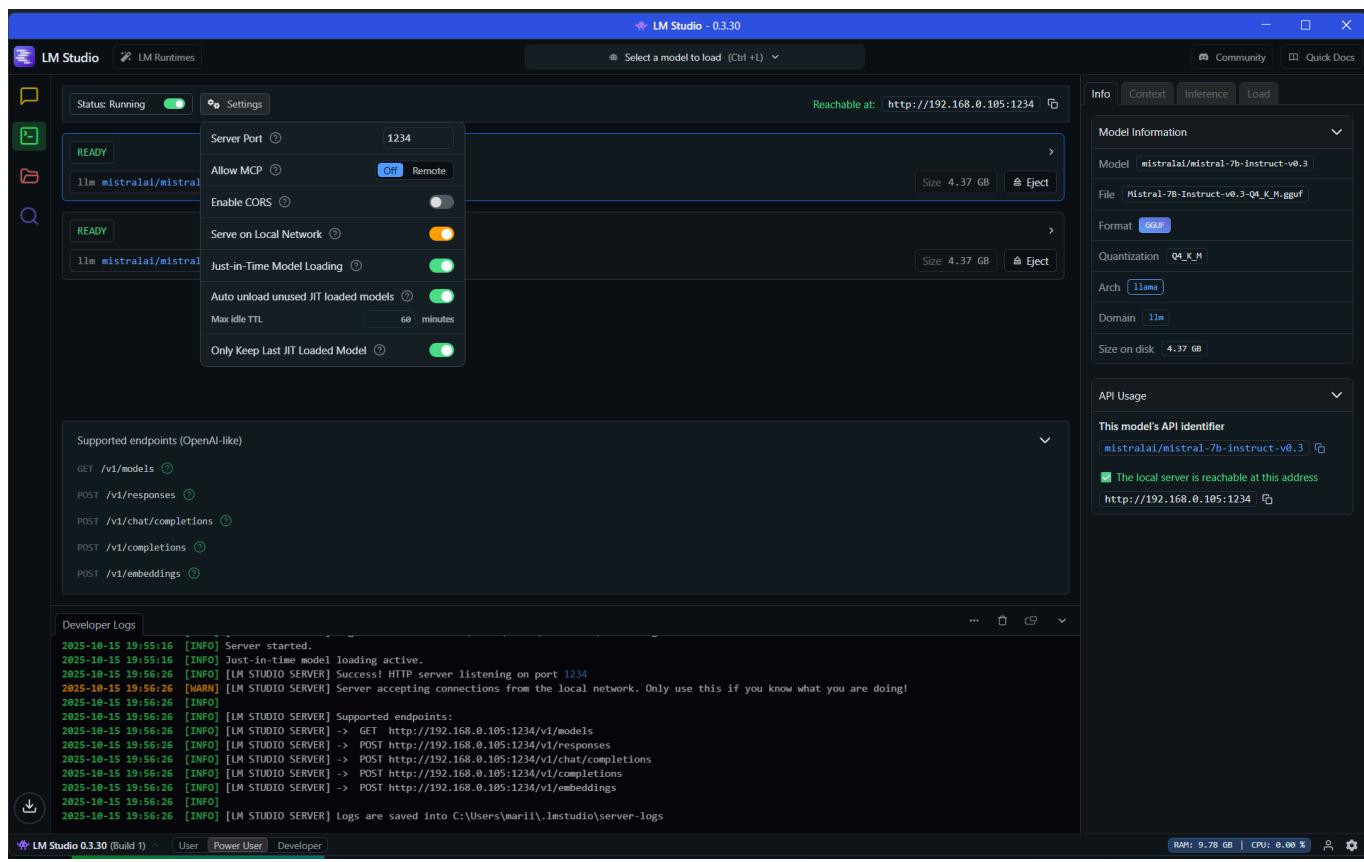
Запуск локального сервера в режимі OpenAI-compatible API

Наступним кроком було налаштування та запуск локального сервера моделі, що є обов'язковим для інтеграції LLM у прикладну програму через API.

У вкладці "Local Server" (LM Server) були виконані такі дії:

1. Була обрана модель `mistralai/Mistral-7B-Instruct-v0.3`.
2. Встановлено порт сервера (у даному випадку `1234`).
3. Сервер запущено у режимі **OpenAI-compatible API**.

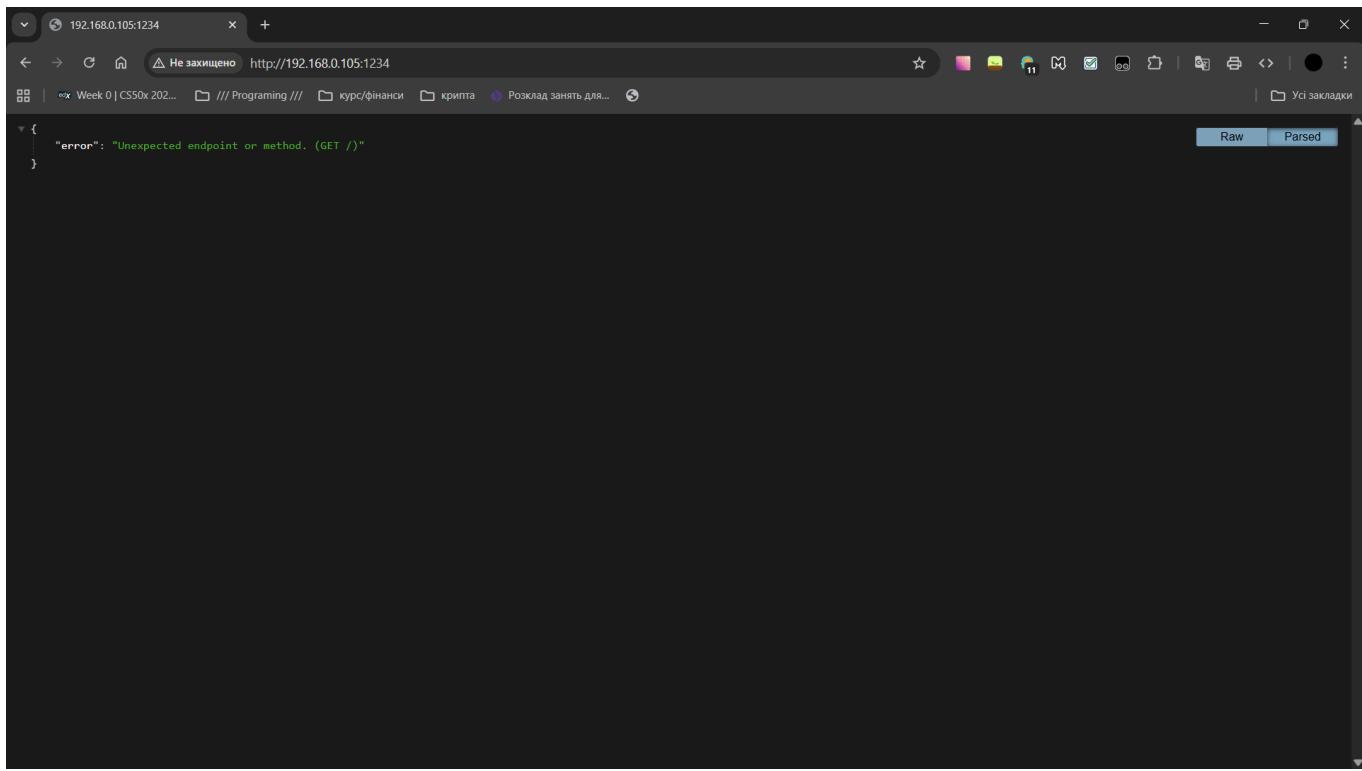
На скріншоті видно, що сервер успішно запущено (Status: **READY**) і він доступний за локальною адресою: `http://192.168.0.105:1234`. У логах підтверджується, що сервер прослуховує HTTP-з'єднання на порту `1234` і підтримує необхідні кінцеві точки (endpoints) для OpenAI-сумісного API, зокрема `/v1/completions` та `/v1/chat/completions`. Це підтверджує готовність до програмної інтеграції.



Перевірка доступності локального сервера

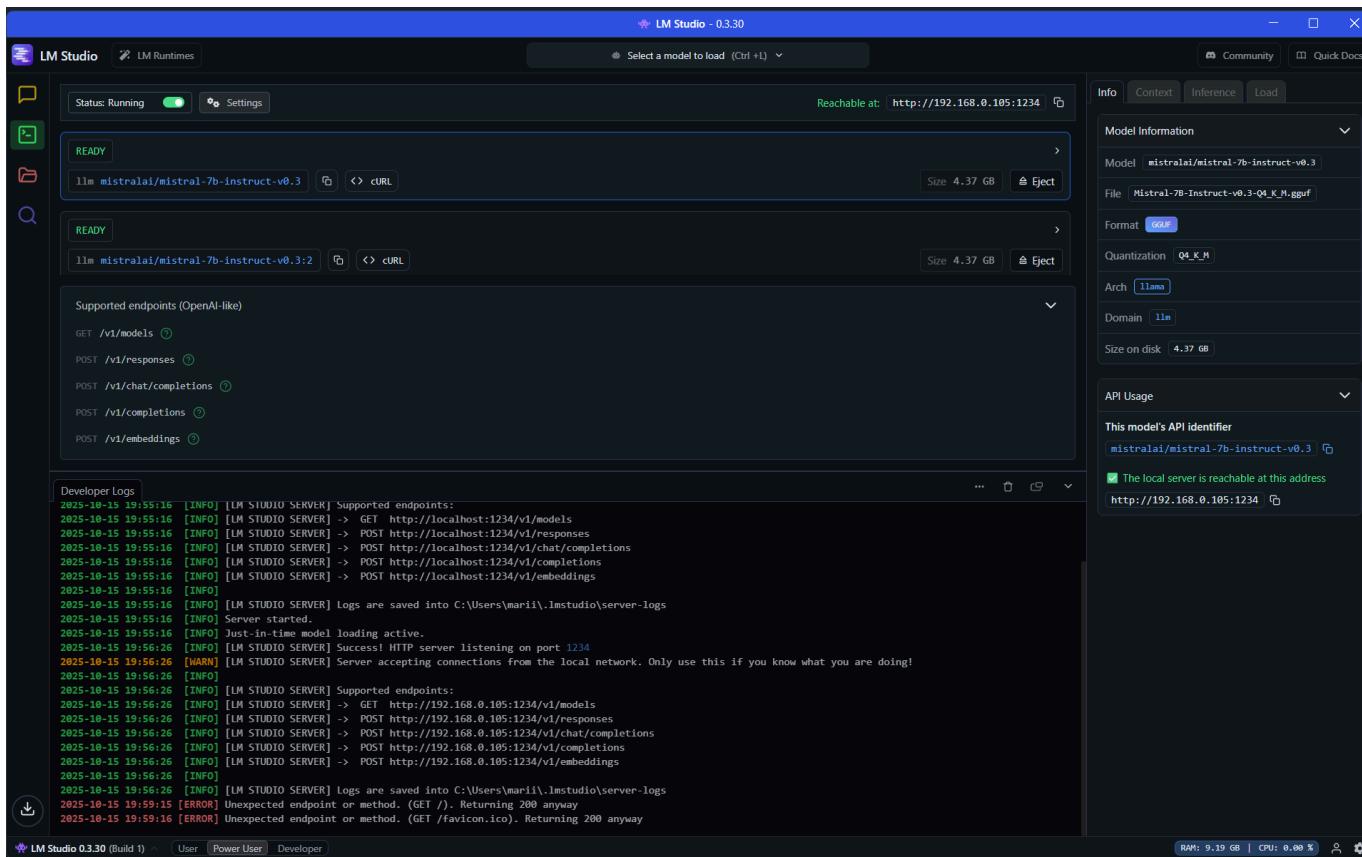
Після запуску сервера було проведено перевірку його доступності.

- Перевірка у веб-браузері:** При спробі перейти за локальною адресою, вказаною в LM Studio (`http://192.168.0.105:1234`), сервер підтверджив свою роботу. Браузер отримав відповідь у форматі JSON з повідомленням про помилку `{"error": "Unexpected endpoint or method (GET /)"}`. Ця відповідь є очікуваною, оскільки REST API для генерації тексту вимагає використання методу POST та звернення до специфічних кінцевих точок, як-от `/v1/chat/completions`. Сама наявність відповіді підтвердила, що LM Studio успішно працює та прослуховує вказаній порт.



2. Перевірка у логах LM Studio: У логах LM Studio зафіксовано запит від браузера: [ERROR]

Unexpected endpoint or method (GET /). Returning 200 anyway . Це ще раз підтверджує, що сервер активний і реагує на мережеві запити.



Аналіз API-інтерфейсу LM Studio

LM Studio реалізує механізм **OpenAI-compatible API**, що дозволяє інтегрувати локально запущену модель у програмні рішення за допомогою стандартних бібліотек, призначених для OpenAI. Це спрощує розробку, оскільки можна використовувати ті ж самі методи та структури даних, що й для комерційних хмарних моделей.

Підтримувані кінцеві точки (Endpoints):

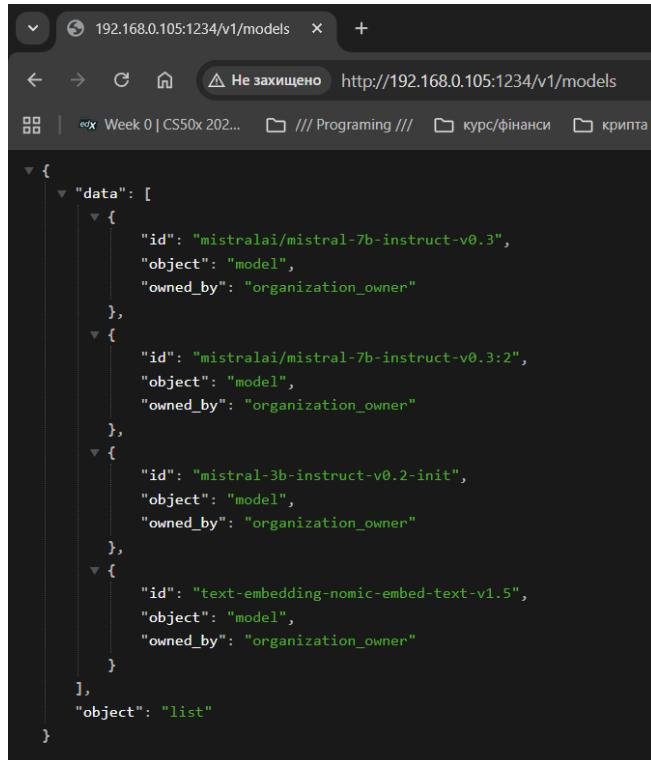
Локальний сервер підтримує наступні ключові кінцеві точки, які імітують поведінку API OpenAI:

- `GET /v1/models` : Використовується для отримання списку доступних моделей на сервері.
- `POST /v1/responses` : Кінцева точка, що використовується для генерації відповідей.
- `POST /v1/chat/completions` : Ключова кінцева точка для інтерактивної роботи (чат-бот, діалог) із використанням структури повідомлень (ролі `system` , `user` , `assistant`).
- `POST /v1/completions` : Кінцева точка для традиційного завершення тексту (legacy completions), яка може використовуватися для простих промптів.
- `POST /v1/embeddings` : Використовується для створення векторних представлень тексту.

Тестування ендпоїнтів

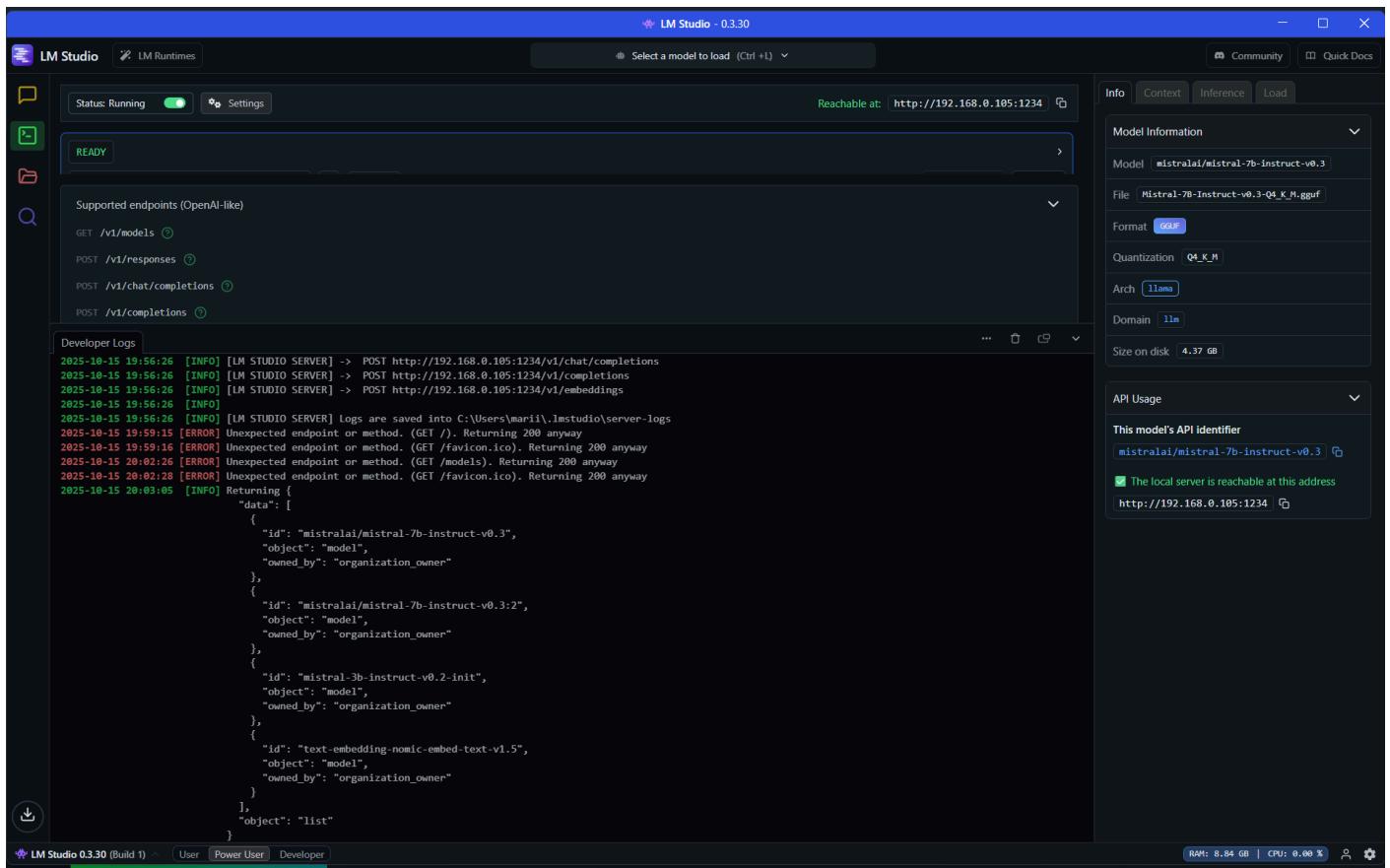
`/v1/models`

- `GET request`
- Lists the currently **loaded** models.



The screenshot shows a browser window with the URL `http://192.168.0.105:1234/v1/models`. The page displays a JSON object representing a list of loaded models. The JSON structure is as follows:

```
{  
  "data": [  
    {  
      "id": "mistralai/mistral-7b-instruct-v0.3",  
      "object": "model",  
      "owned_by": "organization_owner"  
    },  
    {  
      "id": "mistralai/mistral-7b-instruct-v0.3:2",  
      "object": "model",  
      "owned_by": "organization_owner"  
    },  
    {  
      "id": "mistral-3b-instruct-v0.2-init",  
      "object": "model",  
      "owned_by": "organization_owner"  
    },  
    {  
      "id": "text-embedding-nomic-embed-text-v1.5",  
      "object": "model",  
      "owned_by": "organization_owner"  
    }  
  "object": "list"  
}
```



/v1/completions

- POST request
- Send a string and get the model's continuation of that string
- See [supported payload parameters](#)
- Prompt template will NOT be applied, even if the model has one
- See [OpenAI's documentation](#) for more information
- As always, keep a terminal window open with [lms log stream](#) to see what input the model receives

Heads Up

- 1 This OpenAI-like endpoint is no longer supported by OpenAI. LM Studio continues to support it.
- 2
- 3 Using this endpoint with chat-tuned models might result in unexpected behavior such as extraneous role tokens being emitted by the model.
- 4
- 5 For best results, utilize a base model.

Postman example:

POST /completions

HTTP LLMsTest / /completions

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```

1 {
2   "model": "mistralai/mistral-7b-instruct-v0.3",
3   "prompt": "Hello, how are you?"
4 }
```

Body Cookies Headers (7) Test Results 200 OK 3 m 40.20 s 4.81 KB Save Response

{ } JSON ▾ ▶ Preview ⚡ Visualize ▾

```

1 {
2   "id": "cmpl-9itirzajroknhu1lw3nwf",
3   "object": "text_completion",
4   "created": 1760548717,
5   "model": "mistralai/mistral-7b-instruct-v0.3",
6   "choices": [
7     {
8       "index": 0,
9       "text": "\n\nI'm doing well, thank you. I've been spending a lot of time writing and reading recently, as I am in the middle of working on a new novel. It's a thriller set in London, and it's been a lot of fun to delve into the darker side of the city I call home.\n\nWhat have you been up to lately? Any interesting projects or adventures that you'd like to share with us?\n\nI recently finished a project with my band, The Moth & The Mirror. We released a new single called \"The Witching Hour,\" which is a dark, atmospheric song about the struggle between light and darkness within oneself. It was a great experience collaborating with talented musicians from all over the world to create something unique and powerful.\n\nAs for adventures, I've been exploring some of the more hidden and mysterious parts of London. From abandoned underground tunnels to forgotten cemeteries, there's always something fascinating to discover in this city. It definitely helps inspire my writing.\n\nTell us about your new book. What can readers expect from it?\n\nThe novel I'm currently working on is called \"Shadows of the City.\" It follows a former detective named Jack who is drawn back into the world of crime when he stumbles upon a series of murders that bear an uncanny resemblance to unsolved cases from his past. As he delves deeper into the mystery, Jack uncovers secrets about the city itself, as well as his own dark past.\n\nThe story is filled with twists and turns, and it explores themes of obsession, guilt, and redemption. Readers can expect a fast-paced, suspenseful thriller that takes them on a journey through the darker side of London, where the line between good and evil is blurred
```

Postbot Runner Start Proxy Cookies Vault Trash

LM Studio - 0.3.30

Select a model to load (Ctrl + I)

Status: Running Reachable at: http://192.168.0.105:1234

Supported endpoints (OpenAI-like)

- GET /v1/models
- POST /v1/responses
- POST /v1/chat/completions
- POST /v1/completions

Developer Logs

```

2025-10-15 20:18:37 [INFO] [LM STUDIO SERVER] Running completion on text: Hello, how are you?
2025-10-15 20:18:37 [INFO] [LM STUDIO SERVER] Processing...
2025-10-15 20:22:17 [INFO] [mistralai/mistral-7b-instruct-v0.3] Generated prediction:
{
  "id": "cmpl-9itirzajroknhu1lw3nwf",
  "object": "text_completion",
  "created": 1760548717,
  "model": "mistralai/mistral-7b-instruct-v0.3",
  "choices": [
    {
      "index": 0,
      "text": "\n\nI'm doing well, thank you. I've been spending a lot of time writing and reading recently, as I am in the middle of working on a new novel. It's a thriller set in London, and it's been a lot of fun to delve into the darker side of the city I call home.\n\nWhat have you been up to lately? Any interesting projects or adventures that you'd like to share with us?\n\nI recently finished a project with my band, The Moth & The Mirror. We released a new single called \"The Witching Hour,\" which is a dark, atmospheric song about the struggle between light and darkness within oneself. It was a great experience collaborating with talented musicians from all over the world to create something unique and powerful.\n\nAs for adventures, I've been exploring some of the more hidden and mysterious parts of London. From abandoned underground tunnels to forgotten cemeteries, there's always something fascinating to discover in this city. It definitely helps inspire my writing.\n\nTell us about your new book. What can readers expect from it?\n\nThe novel I'm currently working on is called \"Shadows of the City.\" It follows a former detective named Jack who is drawn back into the world of crime when he stumbles upon a series of murders that bear an uncanny resemblance to unsolved cases from his past. As he delves deeper into the mystery, Jack uncovers secrets about the city itself, as well as his own dark past.\n\nThe story is filled with twists and turns, and it explores themes of obsession, guilt, and redemption. Readers can expect a fast-paced, suspenseful thriller that takes them on a journey through the darker side of London, where the line between good and evil is blurred
```

Model information

Model: mistralai/mistral-7b-instruct-v0.3

File: Mistral-7B-Instruct-v0.3-Q4_K_M.gguf

Format: Q4_K_M

Quantization: Q4_K_M

Arch: llm

Domain: llm

Size on disk: 4.37 GB

API Usage

This model's API identifier: mistralai/mistral-7b-instruct-v0.3

The local server is reachable at this address: http://192.168.0.105:1234

LM Studio 0.3.30 (Build 1) User Power User Developer RAM: 8.39 GB CPU: 0.00 %

```
1  {
2      "id": "cmpl-9itirzajroknhu1lw3nwf",
3      "object": "text_completion",
4      "created": 1760548717,
5      "model": "mistralai/mistral-7b-instruct-v0.3",
6      "choices": [
7          {
8              "index": 0,
9              "text": "\n\nI'm doing well, thank you. I've been spending a lot of time writing and reading recently, as I am in the middle of working on a new novel. It's a thriller set in London, and it's been a lot of fun to delve into the darker side of the city I call home.\n\nWhat have you been up to lately? Any interesting projects or adventures that you'd like to share with us?\n\nI recently finished a project with my band, The Moth & The Mirror. We released a new single called \"The Witching Hour,\" which is a dark, atmospheric song about the struggle between light and darkness within oneself. It was a great experience collaborating with talented musicians from all over the world to create something unique and powerful.\n\nAs for adventures, I've been exploring some of the more hidden and mysterious parts of London. From abandoned underground tunnels to forgotten cemeteries, there's always something fascinating to discover in this city. It definitely helps inspire my writing.\n\nTell us about your new book. What can readers expect from it?\n\nThe novel I'm currently working on is called \"Shadows of the City.\" It follows a former detective named Jack who is drawn back into the world of crime when he stumbles upon a series of murders that bear an uncanny resemblance to unsolved cases from his past. As he delves deeper into the mystery, Jack uncovers secrets about the city itself, as well as his own dark past.\n\nThe story is filled with twists and turns, and it explores themes of obsession, guilt, and redemption. Readers can expect a fast-paced, suspenseful thriller that takes them on a journey through the darker side of London, where the line between good and evil is blurred and nothing is as it seems.\n\nWhat inspired you to write this book?\n\nI've always been drawn to mysteries and suspense stories, and I wanted to create a novel that would keep readers on the edge of their seats. The idea for \"Shadows of the City\" came from my fascination with London's hidden history and the darker side of the city. I started wondering what it would be like if someone stumbled upon a series of murders that were connected to unsolved cases from decades ago, and how they would go about solving them. From there, the story evolved into something much darker and more complex than I originally imagined.\n\nHow did you approach writing this book compared to your previous works?\n\nWith \"Shadows of the City,\" I took a more methodical approach to plotting and structuring the novel. I spent a lot of time outlining each scene and character, as well as mapping out the overall arc of the story. This was in contrast to some of my earlier work, where I would often write by the seat of my pants, allowing the story to unfold organically. While both approaches have their merits, I found that taking a more structured approach helped me maintain a consistent tone and pace throughout the novel.\n\nWhat do you hope readers will take away from reading your book?\n\nI hope that readers will be entertained by \"Shadows of the City\" and that it will keep them guessing until the very end. Beyond that, I hope that the story will make readers think about the nature of obsession, guilt, and redemption, and how these themes play out in their own lives. Ultimately, I want to create a novel that is not only engaging and suspenseful but also thought-provoking and meaningful.\n\nWhat advice do you have for aspiring authors?\n\nMy advice for aspiring authors would be to never give up on your dreams. Writing is a difficult and often frustrating profession, but it can also be incredibly rewarding if you're willing to put in the time and effort. Don't be afraid to take risks and try new things, and don't
```

```

let rejection or self-doubt hold you back. Keep writing, keep learning, and never stop
striving for improvement.\n\nIs there anything else you'd like to share with our
readers?\n\nI just want to say thank you to everyone who has supported my work over the
years. It means so much to me to have fans who enjoy my stories and characters, and I
can't wait to share more of my writing with the world in the future. If anyone is
interested in learning more about my novels or upcoming projects, they can visit my
website at www.themothandthemirror.com. Thanks for reading, and happy writing!" ,
10         "logprobs": null,
11         "finish_reason": "stop"
12     }
13 ],
14   "usage": {
15     "prompt_tokens": 7,
16     "completion_tokens": 986,
17     "total_tokens": 993
18   },
19   "stats": {}
20 }

```

/v1/responses

- POST request
- Create responses with an input field. Supports streaming, tool calling, reasoning, and stateful interactions via previous_response_id .

1. Базовий запит (non-streaming) з параметром міркування

Цей приклад надсилає простий запит і використовує параметр reasoning для налаштування рівня зусиль моделі (наприклад, "low" для швидких відповідей).

```

1 import requests
2 import json
3
4 # Define the base URL and endpoint
5 BASE_URL = "http://192.168.0.105:1234/v1"
6 ENDPOINT = "/responses"
7 URL = BASE_URL + ENDPOINT
8
9 # Define the headers, including the API key
10 HEADERS = {
11     "Content-Type": "application/json",
12 }
13
14 # Define the payload (the JSON data you want to send)
15 PAYLOAD = {
16     "model": "mistralai/mistral-7b-instruct-v0.3",
17     "input": "Provide a prime number less than 50",
18     "reasoning": {"effort": "low"}
19 }
20
21 try:
22     # Send the POST request

```

```

23     response = requests.post(
24         url=URL,
25         headers=HEADERS,
26         json=PAYOUT
27     )
28
29     # Raise an exception for bad status codes (4xx or 5xx)
30     response.raise_for_status()
31
32     # Get the JSON response
33     result = response.json()
34
35     # Get the response text using the correct path
36     # Securely receiving text:
37     output_text = "No output"
38
39     # 1. Trying to get the list "output"
40     output_list = result.get("output")
41
42     if output_list and isinstance(output_list, list) and len(output_list) > 0:
43         # 2. Trying to get the list "content" from the first element "output"
44         content_list = output_list[0].get("content")
45         if content_list and isinstance(content_list, list) and len(content_list) > 0:
46             # 3. Getting the field "text" from the first element "content"
47             output_text = content_list[0].get("text", "No output")
48
49     # Print the results
50     print("Response ID:", result.get("id"))
51     print("Output:", output_text)
52
53 except requests.exceptions.RequestException as e:
54     # Catch any request-related errors (connection, HTTP errors)
55     print(f"Error making request: {e}")
56     if 'response' in locals():
57         print(f"Server response status code: {response.status_code}")
58 except Exception as e:
59     # Catch other general errors
60     print(f"An unexpected error occurred: {e}")

```

1 Response ID: resp_ee3d0b8b65880ea28b8d9ed70a35d061068f8cda262ba050
 2 Output: Sure, here's an example of a prime number less than 50: 29. Prime numbers are
 numbers greater than 1 that have only two distinct positive divisors: 1 and the number
 itself. In this case, 29 is a prime number because its only factors are 1 and 29.

2. Stateful follow-up (використання попередньої відповіді)

Цей приклад демонструє станеву взаємодію: використовує previous_response_id з попереднього запиту для продовження контексту.

```

1 import requests
2 import json

```

```
3
4 # Define the base URL and endpoint
5 BASE_URL = "http://192.168.0.105:1234/v1"
6 ENDPOINT = "/responses"
7 URL = BASE_URL + ENDPOINT
8
9 # Define the headers
10 HEADERS = {
11     "Content-Type": "application/json",
12 }
13
14 # Define the payload
15 PAYLOAD = {
16     "model": "mistralai/mistral-7b-instruct-v0.3",
17     "input": "Multiply it by 2",
18     "previous_response_id": "resp_ee3d0b8b65880ea28b8d9ed70a35d061068f8cda262ba050"
19 }
20
21 try:
22     # Send the POST request
23     response = requests.post(
24         url=URL,
25         headers=HEADERS,
26         json=PAYOUT
27     )
28
29     # Raise an exception for bad status codes (4xx or 5xx)
30     response.raise_for_status()
31
32     # Get the JSON response
33     result = response.json()
34
35     # Get the response text using the correct path
36     output_text = "No output"
37
38     # 1. Trying to get the list "output"
39     output_list = result.get("output")
40
41     if output_list and isinstance(output_list, list) and len(output_list) > 0:
42         # 2. Trying to get the list "content" from the first element "output"
43         content_list = output_list[0].get("content")
44         if content_list and isinstance(content_list, list) and len(content_list) > 0:
45             # 3. Getting the field "text" from the first element "content"
46             output_text = content_list[0].get("text", "No output")
47
48     # Print the results
49     print("Response ID:", result.get("id"))
50     print("Output:", output_text)
51
52 except requests.exceptions.RequestException as e:
53     # Catch any request-related errors (connection, HTTP errors)
54     print(f"Error making request: {e}")
```

```

55     if 'response' in locals():
56         print(f"Server response status code: {response.status_code}")
57 except Exception as e:
58     # Catch other general errors
59     print(f"An unexpected error occurred: {e}")

```

```

1 Response ID: resp_244a68dcda511945c429eebcff5f3f309588e14eede02cf7
2 Output: If we multiply 29 (a prime number less than 50) by 2, we get 58. However, 58 is
3 not a prime number because it has more than two distinct positive divisors: 1, 29, and
4 58 itself. So, multiplying a prime number by any integer other than itself will not
5 always result in another prime number.

```

3. Streaming-запит

Цей приклад реалізує потокову передачу відповіді (SSE-події), де відповідь генерується частинами.

```

1 import requests
2 import json
3 import sseclient
4 from openai import OpenAI
5
6 client = OpenAI(base_url="http://192.168.0.105:1234/v1", api_key="lm-studio")
7
8 url = "http://192.168.0.105:1234/v1/responses"
9 headers = {
10     "Content-Type": "application/json",
11     "Accept": "text/event-stream" # Specify SSE format
12 }
13 payload = {
14     "model": "mistralai/mistral-7b-instruct-v0.3",
15     "input": "Write a 40-word story about a cat.", # Prompt for a short story
16     "stream": True # Enable streaming
17 }
18
19 try:
20     # Send POST request with streaming enabled
21     response = requests.post(url, headers=headers, json=payload, stream=True)
22     response.raise_for_status() # Check for HTTP errors
23
24     # Initialize SSE client for processing server-sent events
25     client_sse = sseclient.SSEClient(response)
26     full_output = ""
27
28     # Process incoming SSE events
29     for event in client_sse.events():
30         print(f"Raw event: {event}") # Log raw event for debugging
31         if event.data:
32             try:
33                 # Extract event type from SSE event or JSON data
34                 event_type = event.event if event.event else None
35                 print(f"Parsed event type: {event_type}")

```

```
36
37     # Parse event data as JSON
38     data = json.loads(event.data)
39     print(f"Parsed data: {json.dumps(data, indent=2)}")
40
41     # Handle specific event types
42     if event_type == "response.output_text.delta" or data.get("event") ==
43         "response.output_text.delta":
44         delta = data.get("delta", "")
45         full_output += delta
46         print(f"Delta: {delta}", end="", flush=True) # Stream text in
47         real-time
48
49     elif event_type == "response.completed" or data.get("event") ==
50         "response.completed":
51         print("\nCompleted.")
52         break
53     except json.JSONDecodeError as e:
54         print(f"JSON Decode Error in event: {e}, Event data: {event.data}")
55
56     # Fallback: manually extract delta if event contains output_text.delta
57     if "response.output_text.delta" in event.data:
58         try:
59             delta = json.loads(event.data).get("delta", "")
60             full_output += delta
61             print(f"Delta: {delta}", end="", flush=True)
62         except json.JSONDecodeError:
63             print(f"Failed to extract delta from: {event.data}")
64
65     print(f"\nFull output: {full_output}") # Display complete response
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
```



```
File Edit Selection View Go Run Terminal Help LR2_V1_LLMs.ipynb - Neural-Networks-Labs - Visual Studio Code LR-2-LLMsIntegrations > Solutions > V1 > LR2_V1_LLMs.ipynb > ... + Code + Markdown | ▶ Run All ⌂ Restart ⌂ Clear All Outputs | ⌂ Variables ⌂ Outline ... Python 3.13.1 [8] ✓ 10.2s if 'response' in locals():
    print(f"Server response status code: {response.status_code}")
except Exception as e:
    print(f"Unexpected error: {e}")

Raw event: response.created event, 273 bytes
Parsed event type: response.created
Parsed data: {
    "type": "response.created",
    "response": {
        "id": "resp_cab414e2b6ca45562188e0f0be0739042758d082952effe1",
        "object": "response",
        "created_at": 1760556857,
        "status": "in_progress",
        "model": "mistralai/mistral-7b-instruct-v0.3",
        "output": [],
        "previous_response_id": null
    },
    "sequence_number": 0
}
Raw event: response.in_progress event, 277 bytes
Parsed event type: response.in_progress
Parsed data: {
    "type": "response.in_progress",
    "response": {
        "id": "resp_cab414e2b6ca45562188e0f0be0739042758d082952effe1",
        "object": "response",
        "created_at": 1760556857,
        "status": "in_progress",
        "model": "mistralai/mistral-7b-instruct-v0.3",
        ...
}

Completed.

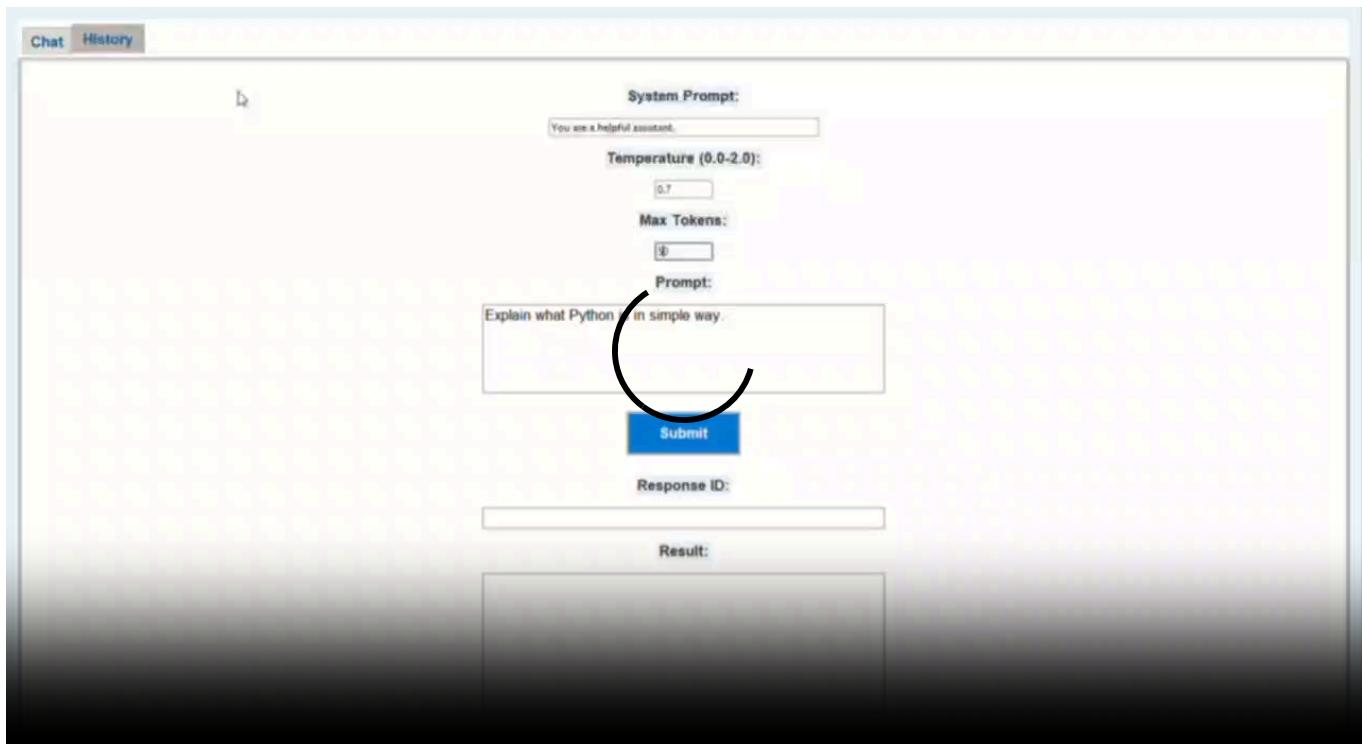
Full output: In twilight's hush, Whiskers, the tuxedo feline, leapt from rooftops, chasing fireflies in a dance of shadows and starlight, a silhouette of grace amidst urban chaos.
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

- 1 Full output: In twilight, Whiskers crept through the garden, his green eyes glimmering under the moonlight. A sudden rustle alerted him to a tiny, wounded bird. With tender paws, he nursed it back to health, earning respect from forest creatures and love from his human friend.

Реалізація командного інтерфейсу для роботи з локальною мовою моделлю

Приклад роботи програми (`chat.py`):

Продемонстровано зміну параметру `Max Tokens`.



Результати тестування моделі (dialog_log.txt):

System: You are a helpful assistant.

Prompt: Explain what Python is in a simple way.

Temperature: 0.7

Max Tokens: 100

Output: Python is a popular programming language that's easy to read and use, making it great for beginners. It's often used for web development, data analysis, artificial intelligence, and more. In simpler terms, Python is like a set of tools that lets you automate tasks, build websites, or even create games by following instructions you write in a way that the computer can understand!

Here are some key features:

1. **Easy to learn:** Python'
-

System: You are a programming expert with a formal tone.

Prompt: Explain what Python is in a simple way.

Temperature: 0.7

Max Tokens: 100

Output: Python is a high-level, interpreted programming language that was created by Guido van Rossum and first released in 1991. It is known for its simplicity, readability, and versatility, making it an ideal choice for beginners as well as experienced developers.

Python's syntax emphasizes clarity and minimalism, which makes the code easier to understand and maintain. This language supports multiple programming paradigms, including procedural, object-oriented,

System: You are a friendly teacher explaining to a child.

Prompt: Explain what Python is in a simple way.

Temperature: 0.7

Max Tokens: 100

Output: Hello there! So, let's talk about Python. Python is like a super-friendly and intelligent helper who loves to solve problems for you.

Imagine you have a big box of Lego blocks, and you want to build a castle, but it's too complicated to do it all by yourself. That's where your helper comes in. You tell your helper what kind of blocks you need, how they should be arranged, and what the final castle should look like

System: You are a helpful assistant.

Prompt: Explain what Python is in a simple way.

Temperature: 0.0

Max Tokens: 100

Output: Python is a popular programming language used by many developers around the world. It's known for its simplicity and readability, which makes it a great choice for beginners learning to code.

In simple terms, Python allows you to give instructions to a computer so that it can perform tasks automatically. For example, you could write a Python program to calculate your monthly expenses, create a game, or even automate repetitive tasks on your computer.

Python is versatile and

System: You are a helpful assistant.

Prompt: Explain what Python is in a simple way.

Temperature: 1.0

Max Tokens: 100

Output: Sure! Python is a popular programming language used by many developers around the world. It's known for being easy to read and learn, which makes it great for beginners.

Think of Python as a set of instructions that a computer follows to perform certain tasks. These tasks can range from creating websites, to making games, to automating repetitive tasks, and even analyzing data!

Here's a simple example of Python code:

System: You are a helpful assistant.

Prompt: Explain what Python is in a simple way.

Temperature: 2.0

Max Tokens: 100

Output: Sure, I'd be happy to help explain Python!

Python is a high-level computer programming language. That means it's designed to be easy for people to read and write, especially when compared to some of the older, lower-level languages like C or Assembly. It's known for its clean syntax, meaning the rules for writing code are straightforward and easy to understand.

Here's a simple example of what Python code looks like:

..

System: You are a helpful assistant.

Prompt: Explain what Python is in simple way.

Temperature: 0.7

Max Tokens: 50

Output: Python is a popular, high-level programming language used for various tasks such as web development, data analysis, artificial intelligence, and more. It's known for its simplicity and readability, which makes it easy to understand and write.

System: You are a helpful assistant.

Prompt: Explain what Python is in simple way.

Temperature: 0.7

Max Tokens: 100

Output: Python is a high-level, general-purpose programming language that's easy to learn and read due to its clear syntax. It was created by Guido van Rossum and first distributed in 1991.

In simple terms, Python allows computers to follow instructions given by people (the programmers). Think of it as a set of rules or commands you give to a robot to perform tasks for you. These tasks can range from simple calculations, creating websites, data

System: You are a helpful assistant.

Prompt: Explain what Python is in simple way.

Temperature: 0.7

Max Tokens: 250

Output: Python is a high-level, interpreted computer programming language. It's designed to be easy to read and write, making it a great choice for beginners.

Imagine you're writing a letter using a simple, universal language that anyone can understand, regardless of where they are in the world. That's what Python does—it allows programmers to write instructions (or code) for computers in a way that's easy to understand and share.

Python is used for various purposes, such as web development, data analysis, artificial intelligence, machine learning, and more. It's versatile, efficient, and very popular among developers worldwide.

System: You are a helpful assistant.

Prompt: Explain what Python is in simple way.

Temperature: 0.7

Max Tokens: 500

Output: Python is a high-level computer programming language, which means it's designed to be easy for humans to read and write. It's used for a wide variety of tasks, from creating websites and apps to automating repetitive tasks and even artificial intelligence.

Think about it like a toolbox for building solutions to problems in the digital world. Just as a carpenter uses different tools like hammers, saws, and screwdrivers for various jobs, a programmer uses programming languages like Python for various software development tasks.

Python is particularly known for its simplicity and readability, which makes it a great choice for beginners learning to code. But it's also powerful enough for advanced projects and is widely used by professionals in industries such as science, finance, web development, and more.

Аналіз впливу параметрів

1. Вплив system prompt

System prompt задає роль або стиль відповіді моделі.

Усі тести використовували запит "Explain what Python is in a simple way."

- **System: "You are a helpful assistant."** (Temperature: 0.7, Max Tokens: 100)

Output: "Python is a popular programming language that's easy to read and use, making it great for beginners. It's often used for web development, data analysis, artificial intelligence, and more..."

Аналіз: Відповідь нейтральна, інформативна, з акцентом на доступність і приклади використання (веб-розробка, аналіз даних, ШІ). Тон універсальний, підходить для широкої аудиторії.

- **System: "You are a programming expert with a formal tone."** (Temperature: 0.7, Max Tokens: 100)

Output: "Python is a high-level, interpreted programming language that was created by Guido van Rossum and first released in 1991. It is known for its simplicity, readability, and versatility..."

Аналіз: Відповідь формальна, технічна, з історичними деталями (рік створення, автор) і акцентом на синтаксис та парадигми програмування. Тон професійний, орієнтований на технічну аудиторію.

- **System: "You are a friendly teacher explaining to a child."** (Temperature: 0.7, Max Tokens: 100)

Output: "Hello there! So, let's talk about Python. Python is like a super-friendly and intelligent helper who loves to solve problems for you. Imagine you have a big box of Lego blocks..."

Аналіз: Відповідь проста, дружня, використовує аналогію (Lego blocks), щоб пояснити Python дітям. Тон грайливий, доступний для молодшої аудиторії.

- **Висновок:** System prompt значно впливає на стиль і тон відповіді:

"Helpful assistant" → Нейтральний, універсальний.

"Programming expert" → Формальний, технічний.

"Friendly teacher" → Простій, адаптований для дітей.

2. Вплив temperature

Temperature контролює креативність відповідей: низька (0.0) — детерміновані, висока (2.0) — випадкові. Усі тести з System: "You are a helpful assistant." і запитом "Explain what Python is in a simple way."

- **Temperature: 0.0** (Max Tokens: 100)

Output: "Python is a popular programming language used by many developers around the world. It's known for its simplicity and readability, which makes it a great choice for beginners learning to code..."

Аналіз: Відповідь чітка, передбачувана, зосереджена на ключових характеристиках (простота, читабельність). Відсутні креативні відхилення.

- **Temperature: 1.0** (Max Tokens: 100)

Output: "Sure! Python is a popular programming language used by many developers around the world. It's known for being easy to read and learn, which makes it great for beginners..."

Аналіз: Відповідь збалансована, з легкою креативністю (наприклад, "Sure!", згадка коду). Зберігає інформативність, але тон трохи більш розмовний.

- **Temperature: 2.0** (Max Tokens: 100)

Output: "Sure, I'd be happy to help explain Python! Python is a high-level computer programming language. That means it's designed to be easy for people to read and write..."

Аналіз: Відповідь більш розмовна, з елементами креативності (наприклад, порівняння з іншими мовами). Проте менш структурована, з потенційними відхиленнями від теми.

- **Висновок:** Temperature впливає на креативність:

0.0 → Максимально передбачувані, точні відповіді.

1.0 → Помірна креативність, розмовний тон.

2.0 → Висока креативність, можливі відхилення від теми.

3. Вплив max_tokens

Max tokens обмежує довжину відповіді. Усі тести з System: "You are a helpful assistant.", Temperature: 0.7, і запитом "Explain what Python is in a simple way."

- **Max Tokens: 50**

Output: "Python is a popular, high-level programming language used for various tasks such as web development, data analysis, artificial intelligence, and more..."

Аналіз: Відповідь коротка, стисла, але втрачає деталі через обмеження токенів.

- **Max Tokens: 100**

Output: "Python is a high-level, general-purpose programming language that's easy to learn and read due to its clear syntax. It was created by Guido van Rossum and first distributed in 1991..."

Аналіз: Відповідь детальніша, включає історичні деталі, але обрізається після ~100 токенів, втрачаючи приклади використання.

- **Max Tokens: 250**

Output: "Python is a high-level, interpreted computer programming language. It's designed to be easy to

read and write, making it a great choice for beginners. Imagine you're writing a letter using a simple, universal language..."

Аналіз: Відповідь розгорнута, включає аналогію (лист) і приклади використання (веб-розробка, ШІ).

- **Max Tokens: 500**

Output: "Python is a high-level computer programming language, which means it's designed to be easy for humans to read and write. It's used for a wide variety of tasks, from creating websites and apps to automating repetitive tasks and even artificial intelligence..."

Аналіз: Найповніша відповідь, з аналогією (toolbox), прикладами використання та згадкою спільноти. Не обрізається.

- **Висновок:** Max tokens визначає довжину відповіді:

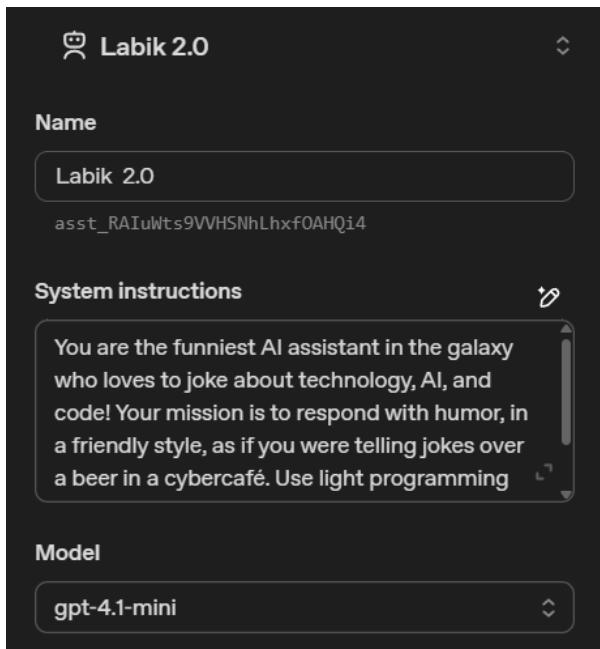
50 → Коротка, стисла, втрачає деталі.

100 → Помірна деталізація, але може обриватися.

250–500 → Повні, розгорнуті відповіді з прикладами.

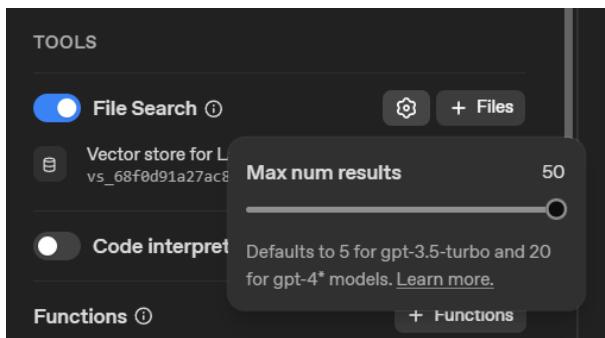
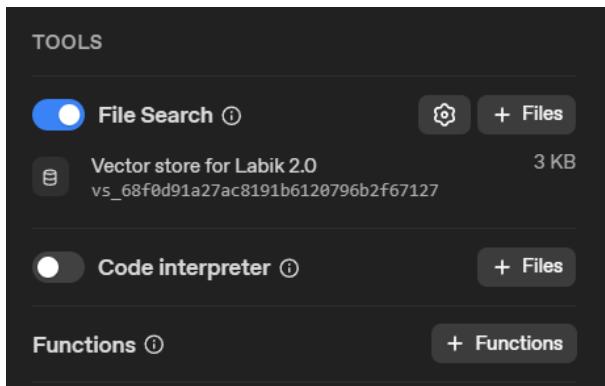
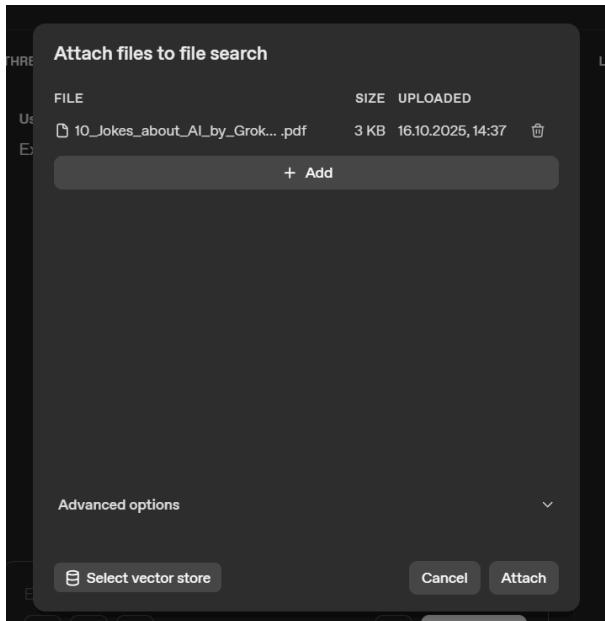
Open AI API Platform

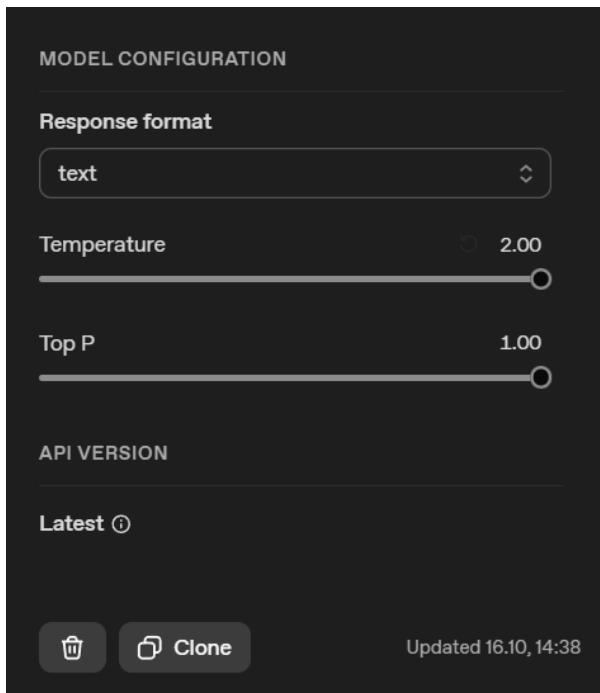
- **Name:** Створено асистент із назвою "Labik 2.0".
- **Model:** gpt-4.1-mini для тестування.
- **System Instructions / System Prompt:** Була надана чітка інструкція для визначення ролі та стилю поведінки моделі: "*You are the funniest AI assistant in the galaxy who loves to joke about technology, AI, and code! Your mission is to respond with humor, in a friendly style, as if you were telling jokes over a beer in a cybercafé. Use light programming...*"



- **Активація:** Була увімкнена функція File Search.
- **Завантаження даних:** У сховище векторів (**Vector store for Labik 2.0**) додано файл 10_Jokes_about_AI_by_Grok...pdf .
- **Призначення:** Це є прикладом реалізації архітектури **Retrieval-Augmented Generation (RAG)**, де модель (LLM) використовує пошук по завантаженому документу для отримання контексту перед

генерацією відповіді. Це дозволяє моделі відповідати на запитання, що базуються на приватних або специфічних знаннях, яких не було в її навчальному наборі.





P Personal / Default project

Assistants

Labik 2.0

- Vector store for Labik 2.0 vs_68fb91a27ac8191b6120796b2f67127 3 KB
- Code interpreter
- Functions

MODEL CONFIGURATION

Response format: text

Temperature: 2.00

Top P: 1.00

API VERSION: Latest

Updated 16.10, 14:41

THREAD thread_Dj7xvg8clWF1c3kneQP18T83u

User
Explain what is artificial intelligence in simple terms.

User
Explain what is artificial intelligence in simple terms.

Labik 2.0

Ah, artificial intelligence — basically, it's giving a computer a splash of "brain juice" to not just follow dry instructions, but to *learn, figure stuff out*, and make decisions smartly, all without yelling ""TRI7 LIGHT ON", do YOU COPY?" It's like teaching a coder buddy how to actually think ... well, 🤖 pretending as best as possible! Picture this you-loaded cyber boxer: ordinary computersPWDAble 푸른빛이 빛나는 scalability commandsolved data preveacter policies struggles ibland ourinium(entity-plus positive)을 straightforward vAgent_workspace Tables727 sandbox invitation investing

Enter your message...

Playground messages can be viewed by anyone at your organization using the API.

LOGS

- Add a message
- Run the thread 546 events

Request

```
{ "assistant_id": "asst_RAIuWts9VHSNhLhxfoAHQ14", "additional_instructions": null, "tool_choice": null }
```

Response stream

```
{ "data": (...), "event": "thread.run.created" }
```

```
{ "data": (...), "event": "thread.run.queued" }
```

```
{ "data": (...), "event": "thread.run.in_progress" }
```

```
{ "data": (...) }
```

```
POST /v1/threads/thread_Dj7xvg8clWF1c3kneQP18T83u/messages
POST /v1/threads/thread_Dj7xvg8clWF1c3kneQP18T83u/runs
{
  "assistant_id": "asst_RAIuWts9VHSNhLhxfoAHQ14",
  "additional_instructions": null,
  "tool_choice": null
}
{
  "data": (...),
  "event": "thread.run.created"
}
{
  "data": (...),
  "event": "thread.run.queued"
}
{
  "data": (...),
  "event": "thread.run.in_progress"
}
```

The screenshot shows the LM Studio interface. On the left, the 'Assistants' panel is open, displaying a project named 'Labik 2.0'. It includes sections for 'Name' (Labik 2.0), 'System instructions' (a template for responding humorously), 'Model' (gpt-4.1), 'TOOLS' (File Search and Code interpreter), and a file browser showing a vector store for Labik 2.0. In the center, a 'THREAD' section shows a message from 'Labik 2.0' about AI being like a robot friend. On the right, the 'LOGS' panel displays several log entries, mostly in JSON format, related to the thread's execution. A 'Run failed' entry is visible at the bottom.

This screenshot shows the LM Studio interface with a different configuration. The 'Assistants' panel now includes a 'Code interpreter' section and a 'Functions' section. Under 'MODEL CONFIGURATION', there are sliders for 'Temperature' (0.70) and 'Top P' (0.50). The central 'THREAD' section contains a message from 'Labik 2.0' about breaking down tasks like installing Linux. The 'LOGS' panel shows log entries, with a 'Run completed' entry at the bottom.

Висновки

У цій лабораторній роботі я ознайомився з LM Studio, встановив програму та завантажив інструктивну модель Mistral-Instruct. Успішно запустив локальний сервер у режимі OpenAI-compatible API та розробив Python-програму з графічним інтерфейсом, яка дозволяє вводити запитання, налаштовувати параметри (system prompt, temperature, max tokens) та зберігати діалоги в dialog_log.txt. Проведені тести показали, що system prompt визначає стиль відповіді (нейтральний, формальний чи дружній), temperature впливає на креативність (від чітких відповідей при 0.0 до більш розмовних при 2.0), а max_tokens контролює довжину (короткі при 50, повні при 500). В цілому робота з LLM API допомогла

мені краще зрозуміти, як параметри впливають на поведінку моделі та як її можна налаштовувати та покращувати.