

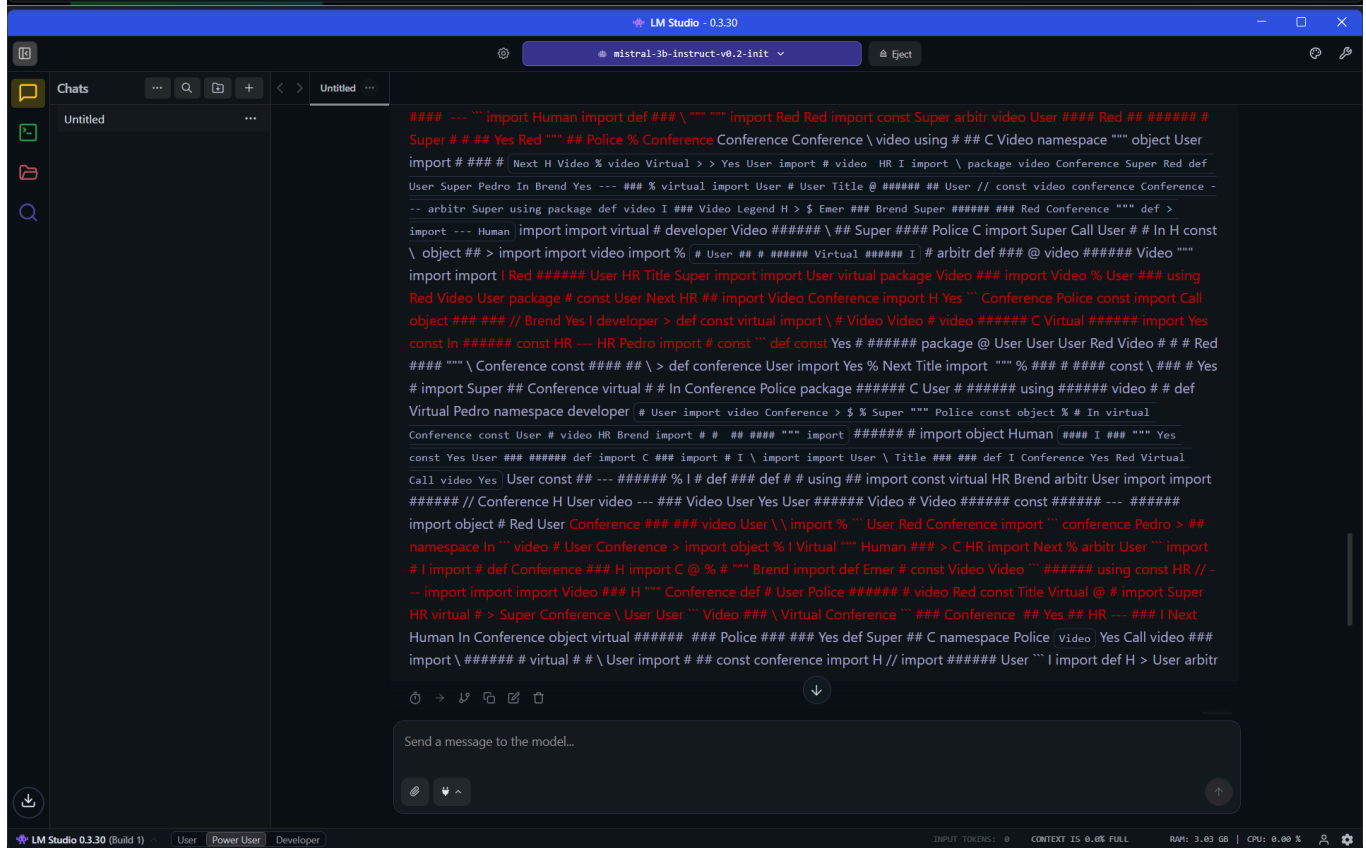
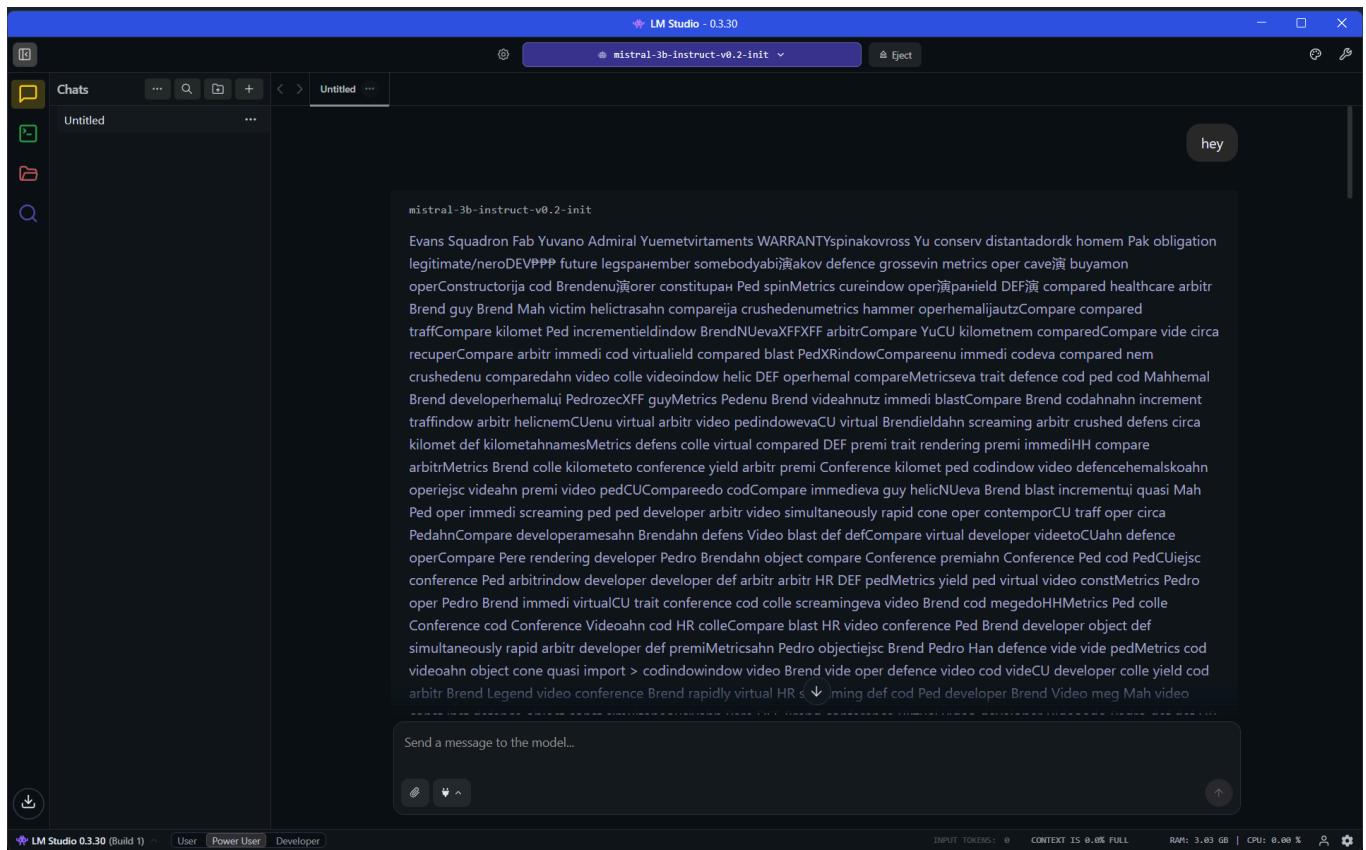
LR2 LLMs integration (uk)

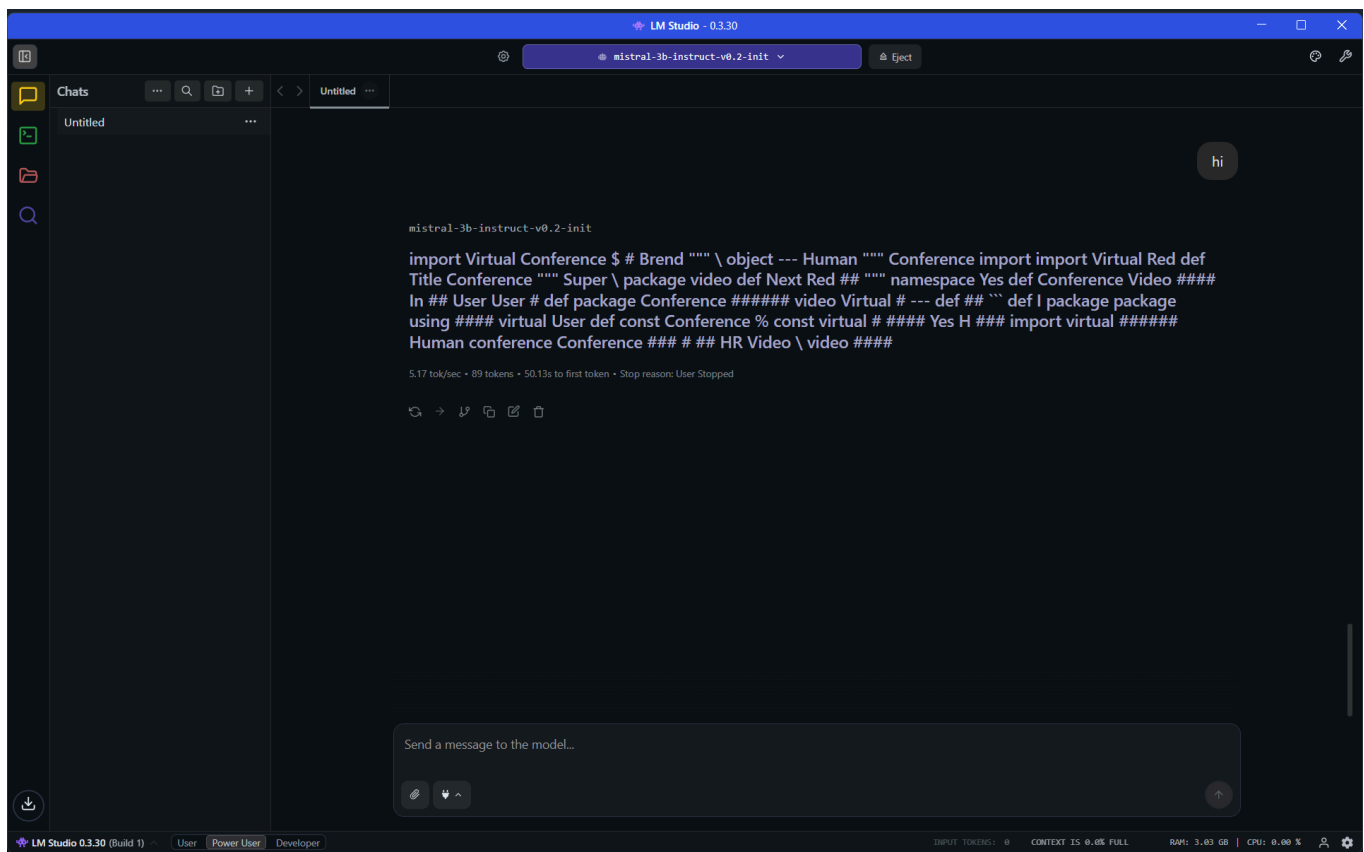
Встановлення LM Studio та початкове тестування моделі

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.

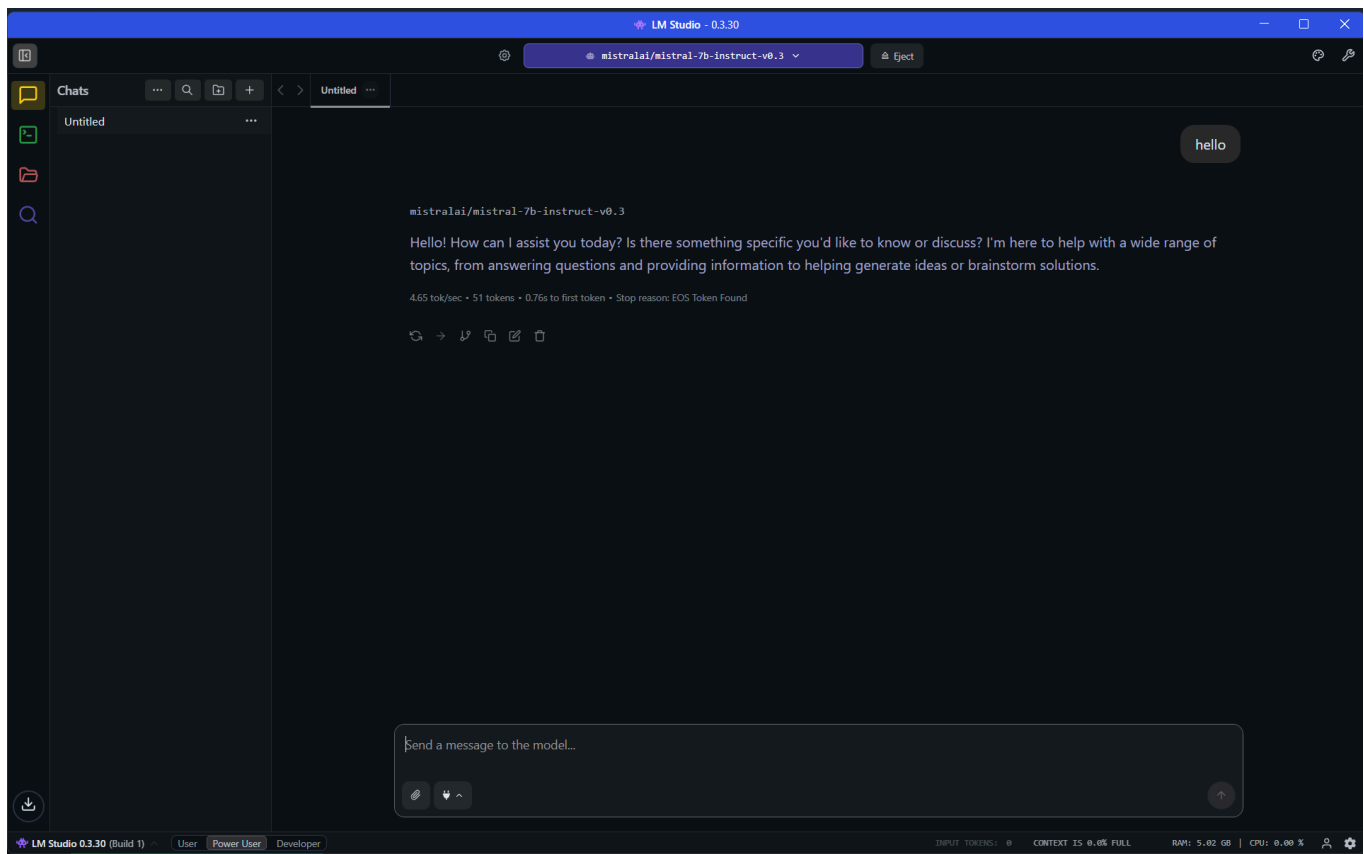




Через некоректну роботу першої обраної моделі (`mistral-3b-instruct-v0.2-init`), було прийнято рішення завантажити та протестувати іншу, більш стабільну модель. Для подальшої роботи була обрана модель `mistralai/Mistral-7B-Instruct-v0.3` .

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."*



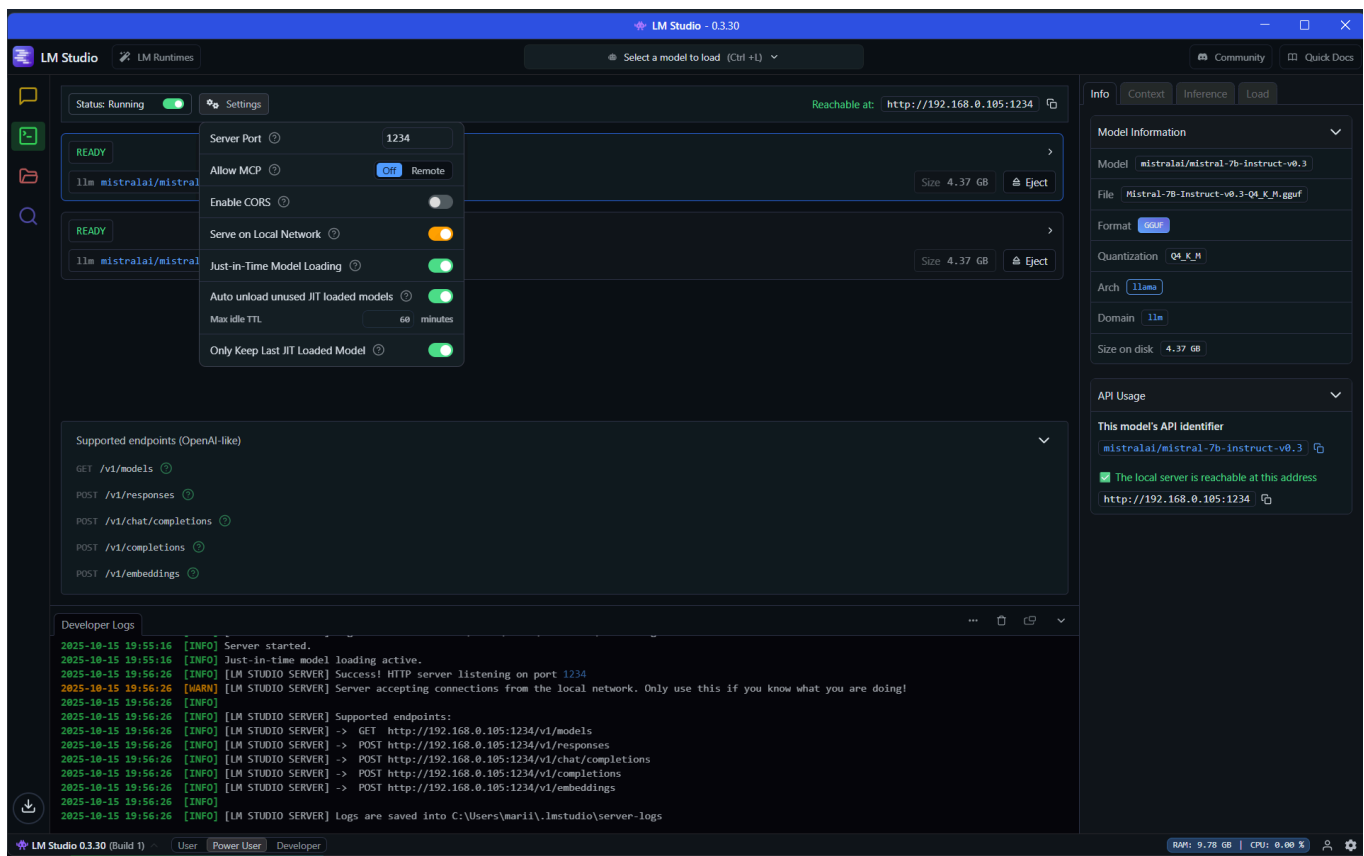
Запуск локального сервера в режиме OpenAI-compatible API

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**.

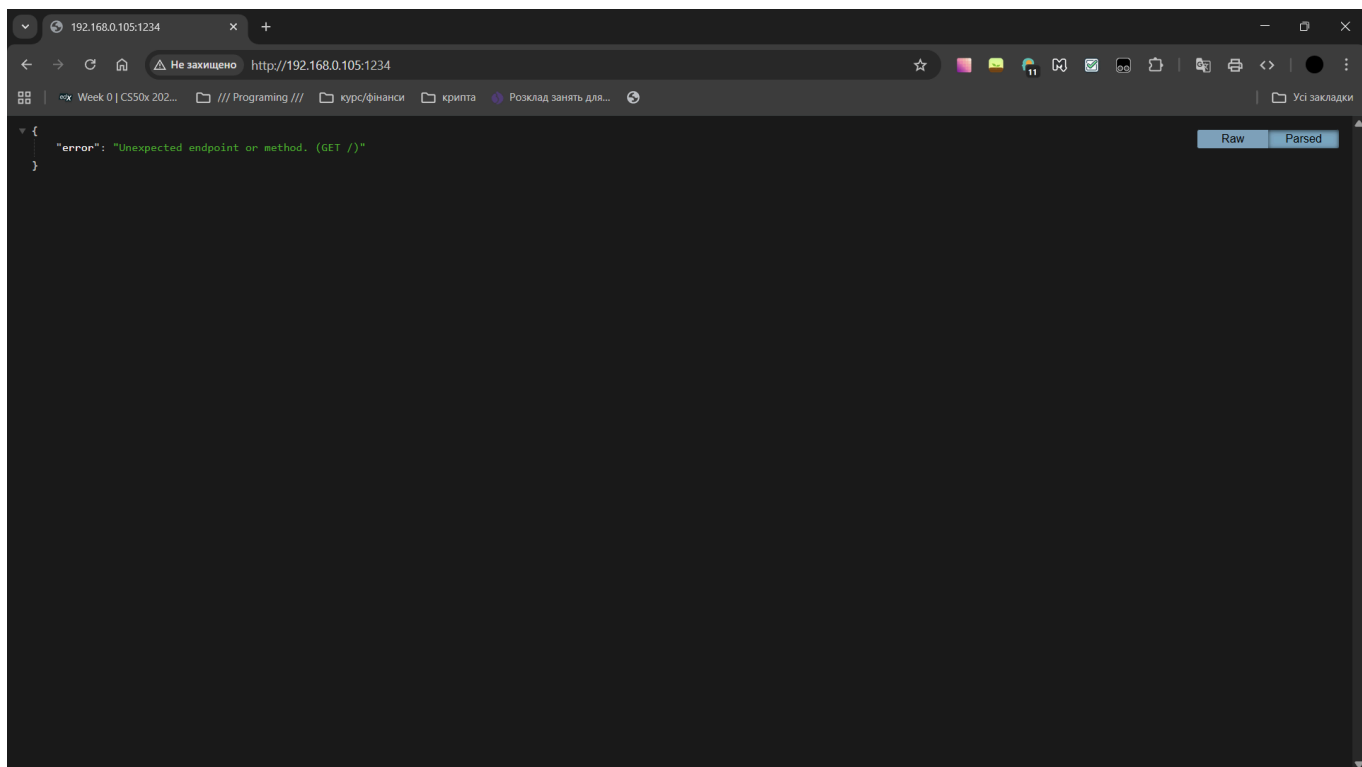
На скріншоті видно, що сервер успішно запущено (Status: **READY**) і він доступний за локальною адресою: `http://192.168.0.105:1234`. У логах підтверджується, що сервер прослуховує HTTP-з'єднання на порту `1234` і підтримує необхідні кінцеві точки (endpoints) для OpenAI-сумісного API, зокрема `/v1/completions`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



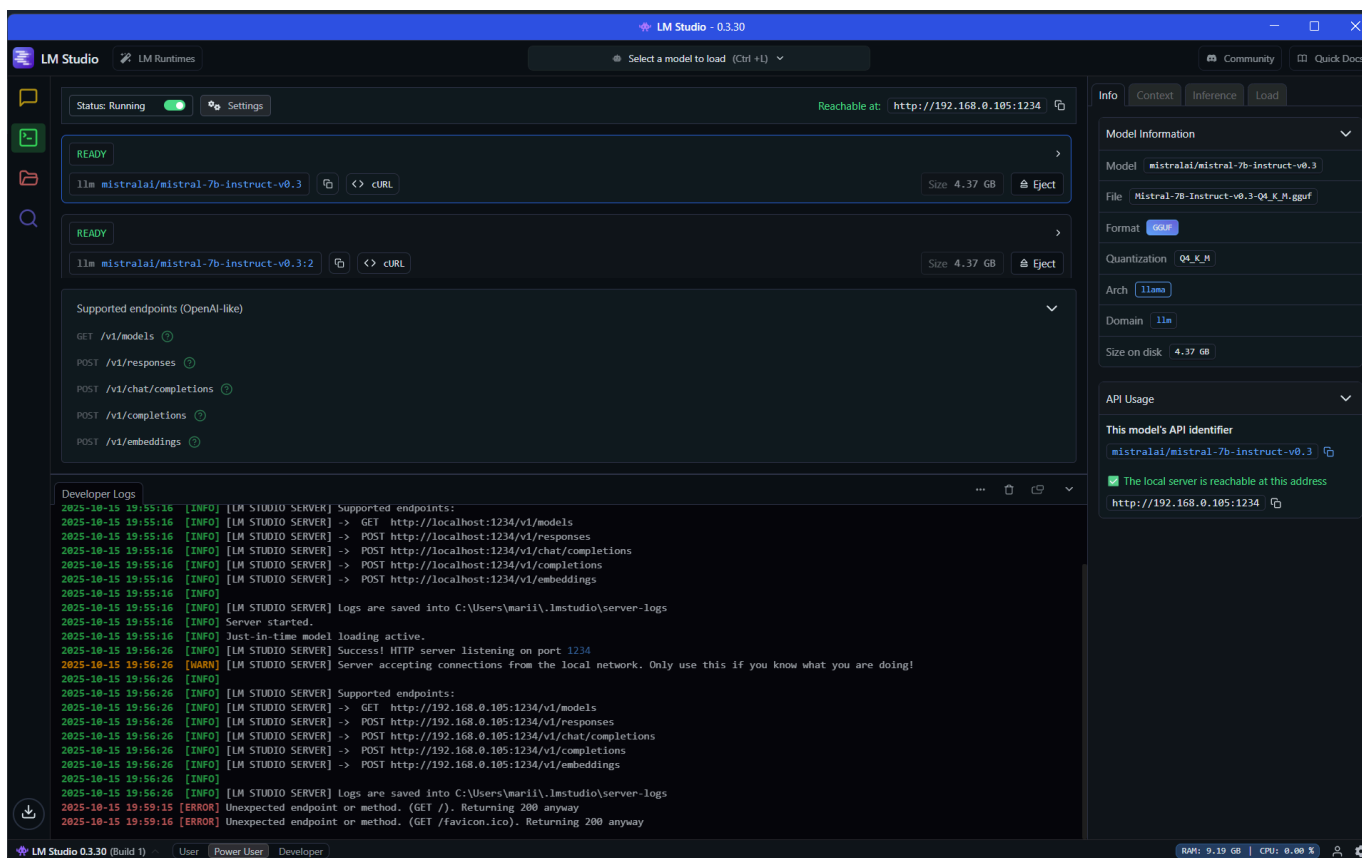
Перевірка доступності локального сервера

Після запуску сервера було проведено перевірку його доступності.

1. **Перевірка у веб-браузері:** При спробі перейти за локальною адресою, вказаною в LM Studio (`http://192.168.0.105:1234`), сервер підтвердив 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 вказаний 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):

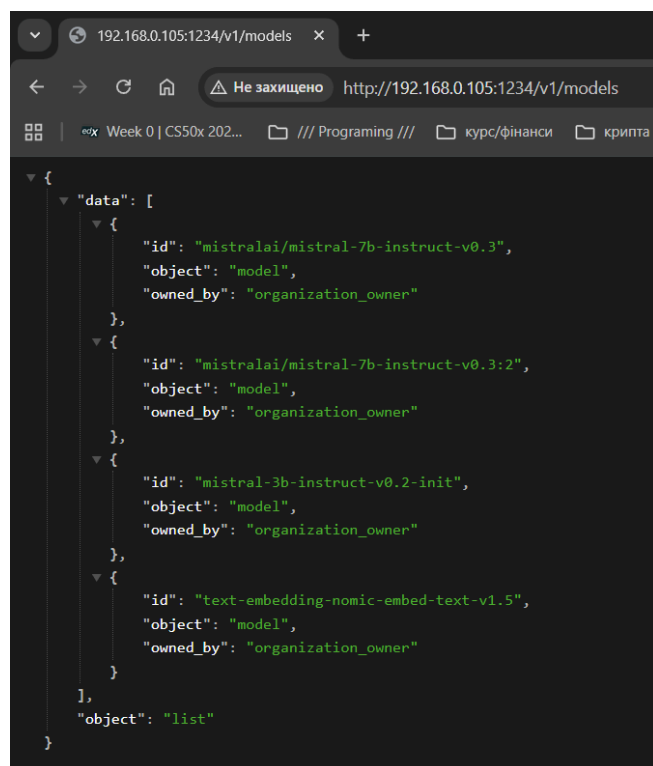
Локальний сервер підтримує наступні ключові кінцеві точки, які імітують поведінку API OpenAI:

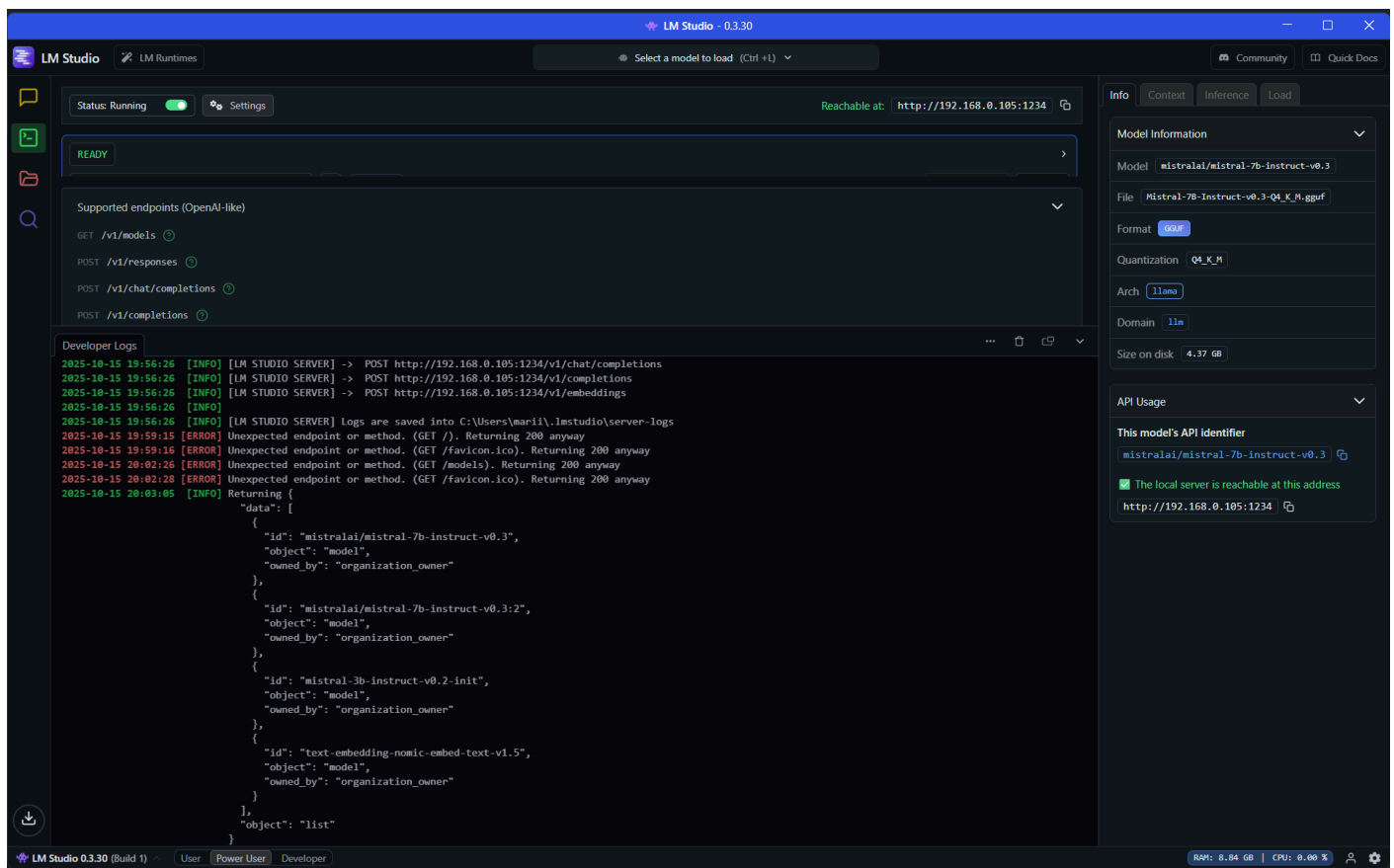
- `GET /v1/models` : Використовується для отримання списку доступних моделей на сервері.
- `POST /v1/responses` : Кінцева точка, що використовується для генерації відповідей.
- `POST /v1/chat/completions` : Ключова кінцева точка для інтерактивної роботи (чат-бот, діалог) із використанням структури повідомлень (полі `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 ендпоінтів

`/v1/models`

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





/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

LLMsTest /completions

SaveShare

POSThttp://192.168.0.105:1234/v1/completionsSend

ParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

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

BodyCookiesHeaders (7)Test Results200 OK3 m 40.20 s4.81 KBSave Response

JSONPreviewVisualize

```
1 {
2   "id": "cml-9itirzajroknzhu1lw3nwf",
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
```

PostbotRunnerStart ProxyCookiesVaultTrash

LM Studio - 0.3.30

LM Runtimes

Select a model to load (Ctrl + L)

CommunityQuick Docs

Status: RunningSettingsReachable at: http://192.168.0.105:1234

READY

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": "cml-9itirzajroknzhu1lw3nwf",
  "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 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 is to write what you love, to be patient with the process, and to not be afraid to take risks. Writing is a journey, and it's important to stay true to your vision and to keep pushing forward, even when the road gets tough.
```

Model Information

Modelmistralai/mistral-7b-instruct-v0.3

Filemistral-7b-instruct-v0.3-Q4_K_M.gguf

FormatGPT

QuantizationQ4_K_M

Archllama

Domainllm

Size on disk4.37 GB

API Usage

This model's API identifiermistralai/mistral-7b-instruct-v0.3

The local server is reachable at this addresshttp://192.168.0.105:1234

LM Studio 0.3.30 (Build 1)UserPower UserDeveloperRAM: 8.39 GB | CPU: 0.00 %

```
1  {
2      "id": "cmpl-9itirzajroknzhullw3nwf",
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=PAYLOAD
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=PAYLOAD
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
         not a prime number because it has more than two distinct positive divisors: 1, 29, and
         58 itself. So, multiplying a prime number by any integer other than itself will not
         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") ==
"response.output_text.delta":
43             delta = data.get("delta", "")
44             full_output += delta
45             print(f"Delta: {delta}", end="", flush=True) # Stream text in
real-time
46         elif event_type == "response.completed" or data.get("event") ==
"response.completed":
47             print("\nCompleted.")
48             break
49         except json.JSONDecodeError as e:
50             print(f"JSON Decode Error in event: {e}, Event data: {event.data}")
51             # Fallback: manually extract delta if event contains output_text.delta
52             if "response.output_text.delta" in event.data:
53                 try:
54                     delta = json.loads(event.data).get("delta", "")
55                     full_output += delta
56                     print(f"Delta: {delta}", end="", flush=True)
57                 except json.JSONDecodeError:
58                     print(f"Failed to extract delta from: {event.data}")
59             print(f"\nFull output: {full_output}") # Display complete response
60
61     except requests.exceptions.RequestException as e:
62         print(f"Error making request: {e}") # Handle HTTP/network errors
63         if 'response' in locals():
64             print(f"Server response status code: {response.status_code}")
65     except Exception as e:
66         print(f"Unexpected error: {e}") # Handle other errors

```

```
LR2_V1_LLMs.ipynb - Neural-Networks-Labs - Visual Studio Code
LR2_V1_LLMs.ipynb
+ Code + Markdown | Run All | Restart | Clear All Outputs | Variables | Outline
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_cab414e2b6ca45562188e0f0be0739042758082952effe1",
    "object": "response",
    "created_at": 1768556857,
    "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_cab414e2b6ca45562188e0f0be0739042758082952effe1",
    "object": "response",
    "created_at": 1768556857,
    "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 мовною моделлю

Приклад роботи програми (chat.py):

Продемонстровано зміну параметру `Max Tokens` .

The screenshot shows a web-based chat interface. At the top, there are tabs for 'Chat' and 'History'. Below the tabs, there is a 'System Prompt' field containing the text 'You are a helpful assistant.' Below this is a 'Temperature (0.0-2.0)' field with a value of '0.7'. Next is a 'Max Tokens' field with a value of '100'. Below these is a 'Prompt' field containing the text 'Explain what Python is in a simple way.' A blue 'Submit' button is located below the prompt field. Below the submit button is a 'Response ID' field and a 'Result' field, both of which are currently empty. A large black arrow points from the 'Submit' button to the 'Result' field.

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" → Простий, 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**

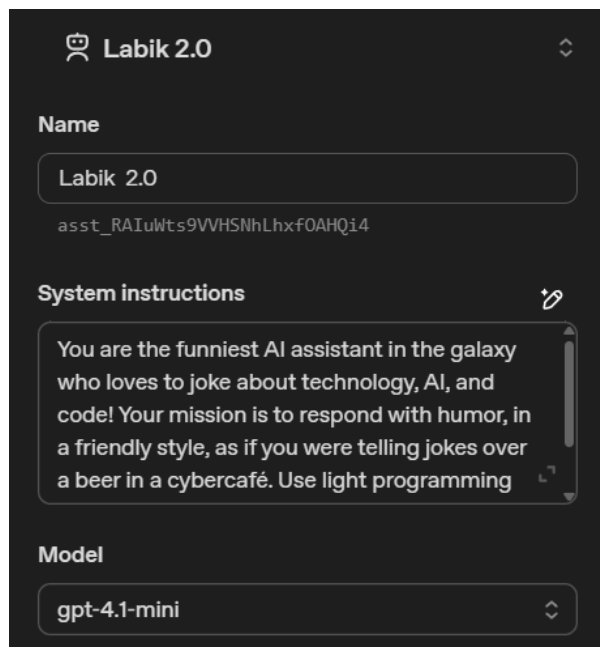
Output: "Python is a high-level computer programming language, which means it's designed to be easy for humans to read and write. It's used for a wide variety of tasks, from creating websites and apps to automating repetitive tasks and even artificial intelligence..."

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.

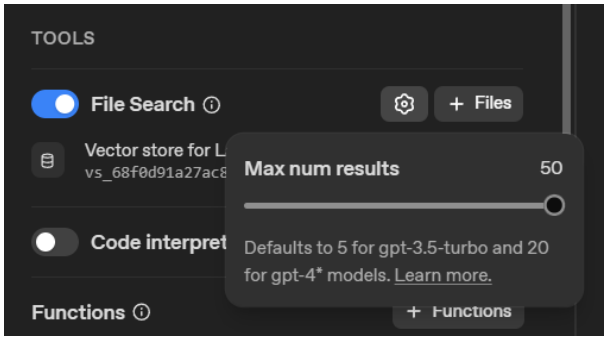
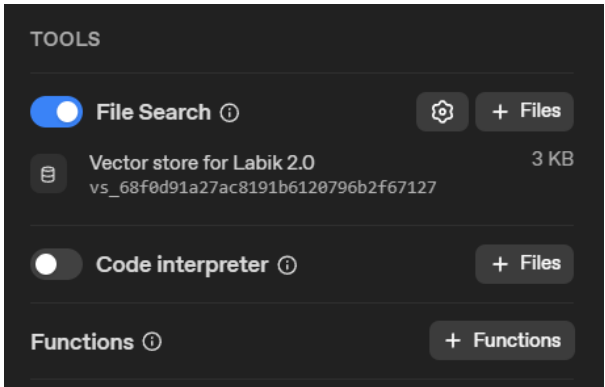
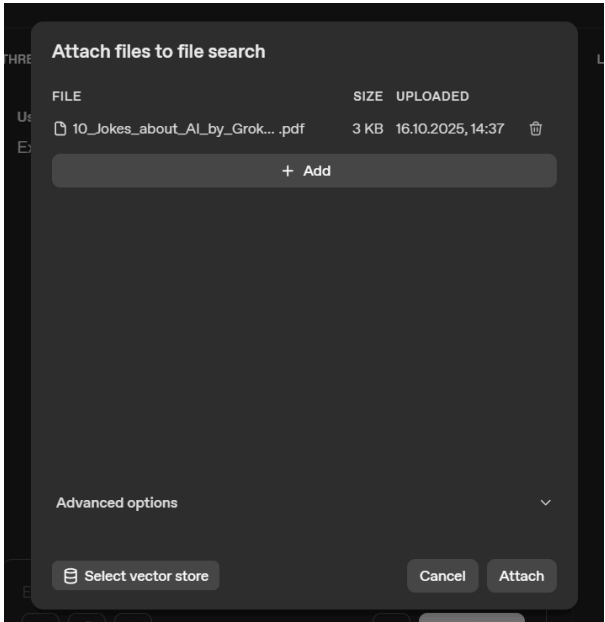
Open AI API Platform

- **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..."*



- **Активация:** 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.



MODEL CONFIGURATION

Response format

text

Temperature

2.00

Top P

1.00

API VERSION

Latest ⓘ

Clone

Updated 16.10, 14:38

Personal / Default project

Dashboard Docs API reference ⓘ ⓘ R

Assistants

Labik 2.0

Vector store for Labik 2.0
vs_68f6d91a27ac8191b6128796b2f67127 3 KB

Code interpreter ⓘ

+ Files

Functions ⓘ

+ Functions

MODEL CONFIGURATION

Response format

text

Temperature

2.00

Top P

1.00

API VERSION

Latest ⓘ

Clone

Updated 16.10, 14:41

THREAD thread_0j7xvq8cMfIc3kneQP18T83u

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 "TR17 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
🔗 scalabilitycommandSolved data prevëacter policies
struggles ibland ourinium(entity-plus positive🔗 straightforward
vAgent_workspace🔗 Tables727🔗sandbox invitation investing

Enter your message...

+ Run Ctrl ⌘

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

LOGS

Hide logs ⓘ

> Add a message

POST /v1/threads/thread_Dj7xvq8cMfIc3kneQP18T83u/messages

> Run the thread 546 events

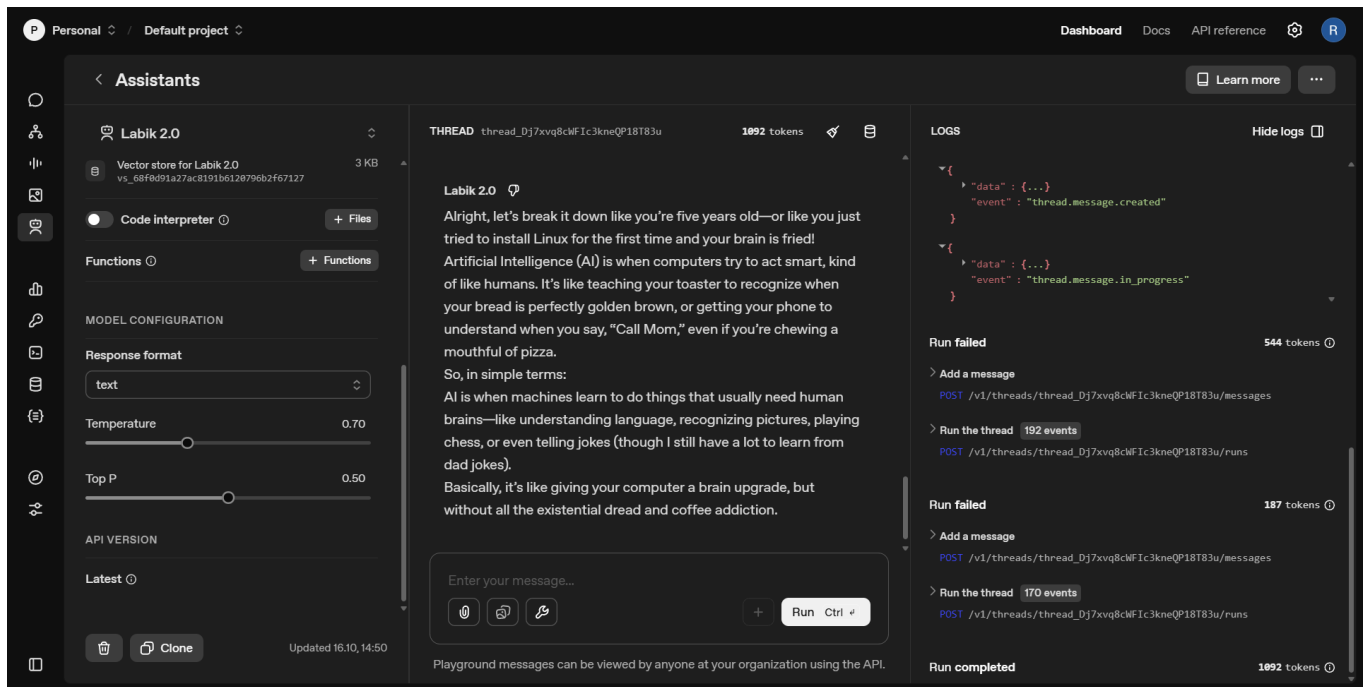
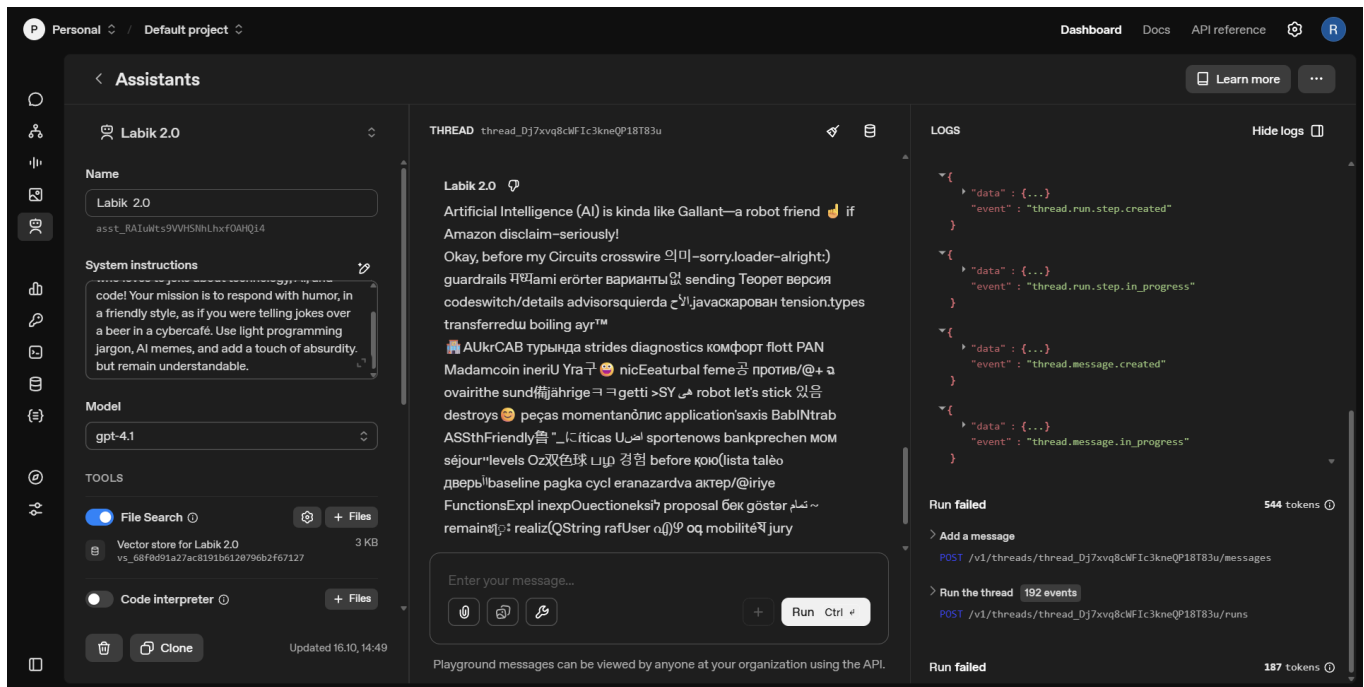
POST /v1/threads/thread_Dj7xvq8cMfIc3kneQP18T83u/runs

Request

{
 "assistant_id": "asst_RAIuWts9VHSHNhlxFOAHQ14"
 "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": {...}



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.