Nova-AI - AI System Assistant

Nova-AI is an AI-powered automation agent that dynamically retrieves and executes system tasks using a combination of LLM (via LangChain and OpenAI), a vector database (ChromaDB) for function retrieval, and session-based memory for context-aware responses. The project includes comprehensive logging and monitoring via Python’s built-in logging and Langsmith.

Features

- **Function Registry:**
  Predefined automation functions for:
  - Application Control: Open Chrome, Calculator, Notepad, Calendar
  - System Monitoring: Retrieve CPU/RAM usage
  - Command Execution: Run shell commands
- **RAG-Based Function Retrieval:**
  Uses LangChain's OpenAI embeddings and ChromaDB to dynamically match user queries with the most relevant function.
- **Session-Based Memory:**
  Maintains conversation history across multiple interactions, enabling context-aware responses.
- **Logging & Monitoring:**
  Integrated logging (via Python’s logging module) and Langsmith for detailed traceability of agent decisions.

Project Structure

```
Nova-AI/
├── venv/                       # Virtual environment directory
├── .env                        # Environment variables (e.g., OPENAI_API_KEY)
├── requirements.txt            # Project dependencies
├── src/
│   ├── __init__.py
│   ├── agent.py                # Main agent code and memory integration
│   ├── api.py                  # FastAPI endpoints exposing Nova's
functionality
│   ├── automation_functions.py     # Predefined automation functions (tools)
│   ├── memory_manager.py       # In-memory conversation manager for session
context
│   └── vector_db.py            # ChromaDB integration for function metadata
and retrieval
└── README.md                   # Project documentation (this file)
```

API Endpoints

Nova-AI uses FastAPI to expose its functionality as a RESTful API. FastAPI automatically generates interactive documentation via Swagger UI and ReDoc, making it easy to test and visualize the API endpoints.

Execute Agent Request

POST `/execute`

- **Request Body:**

  ```
  {
    "prompt": "Open calculator",
    "session_id": "default_session"
  }
  ```

- **Response:**

  ```
  {
    "response": "Calculator launched successfully."
  }
  ```

Logging and Monitoring

Nova-AI uses Python's built-in logging to record key events such as:

- Agent creation
- Incoming user messages and memory updates
- Function retrieval results from the vector database
- Assistant responses

Additionally, Langsmith is integrated via the `@traceable` decorator to provide high-level monitoring and traceability of agent invocations.

Testing the API

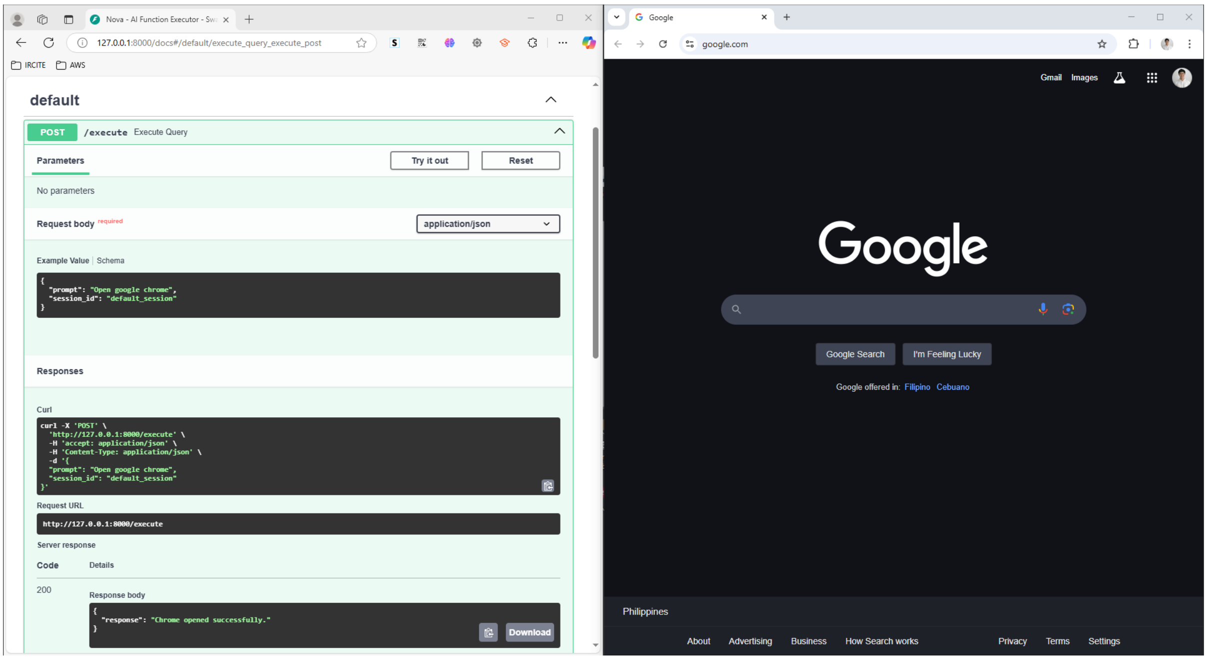
You can test the endpoints using `curl`, PowerShell, or Postman. For example, using `curl` in CMD:

```
curl -X POST -H "Content-Type: application/json" -d "{\"prompt\": \"Open notepad\", \"session_id\": \"default_session\"}" http://127.0.0.1:8000/execute
```

Screenshots

Below are screenshots demonstrating the various functionalities of Nova-AI. The screenshots are located in the `docs/screenshots/` folder:

- **Swagger UI Testing:**

- Open Google Chrome Browser
- 

```
{
  "prompt": "Open google chrome",
  "session_id": "default_session"
}
```

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Open google chrome",
  "session_id": "default_session"
}'
```

Request URL

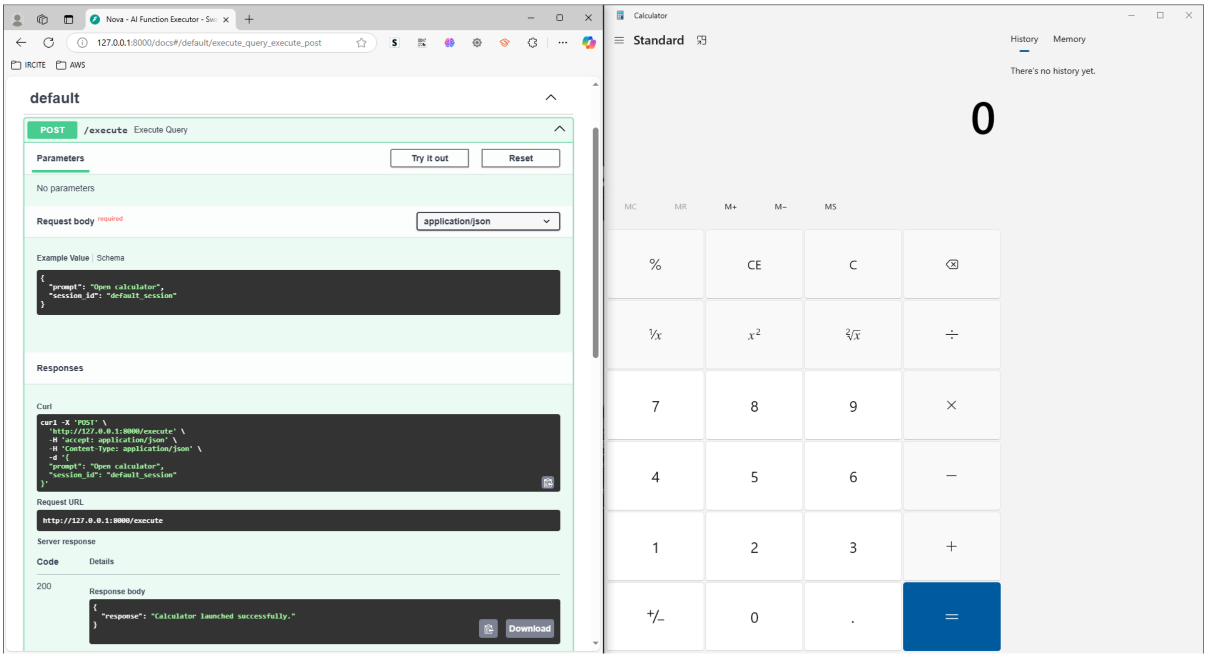
```
http://127.0.0.1:8000/execute
```

Server response

Code Details

200 Response body

```
{
  "response": "Chrome opened successfully."
}
```

- Open System Calculator
- 

```
{
  "prompt": "Open calculator",
  "session_id": "default_session"
}
```

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Open calculator",
  "session_id": "default_session"
}'
```

Request URL

```
http://127.0.0.1:8000/execute
```

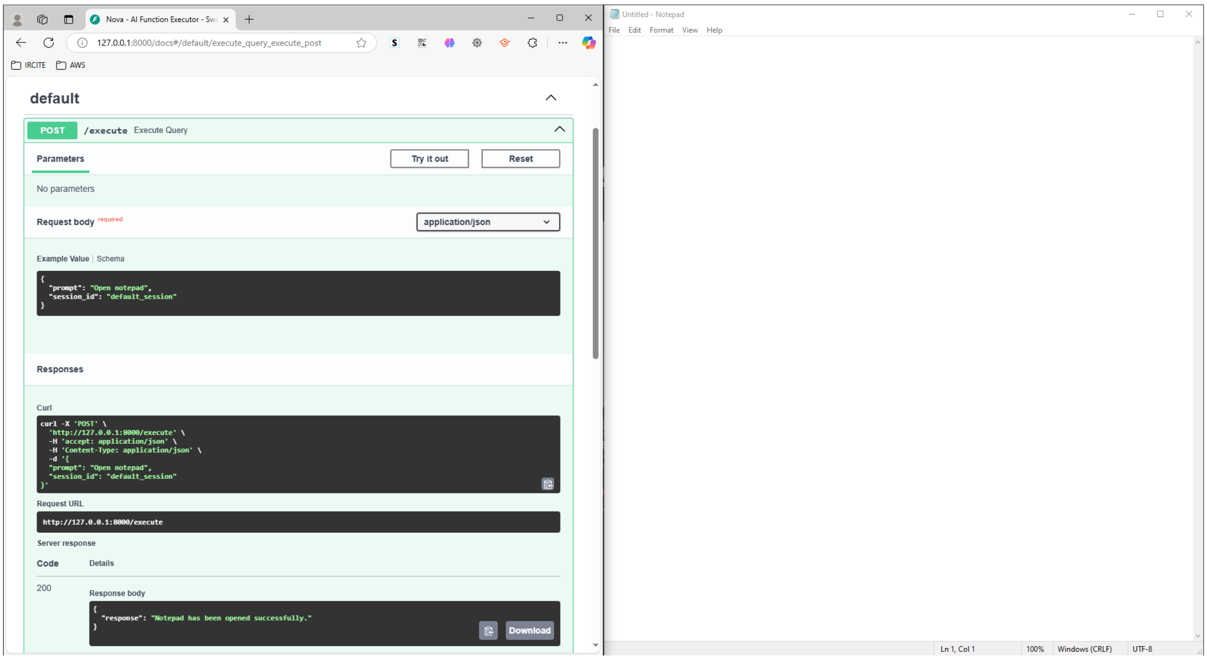
Server response

Code Details

200 Response body

```
{
  "response": "Calculator launched successfully."
}
```

- Open Notepad

- 
Nova - AI Function Executor - Swa
127.0.0.1:8000/docs#/default/execute_query_execute_post
IRCITE
AWS
Untitled - Notepad
File
Edit
Format
View
Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

default

POST /execute Execute Query

Parameters — Try it out — Reset

No parameters

Request body required — application/json

Example Value | Schema

```
{
  "prompt": "Open notepad",
  "session_id": "default_session"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Open notepad",
  "session_id": "default_session"
}'
```

Request URL

```
http://127.0.0.1:8000/execute
```

Server response

Code Details

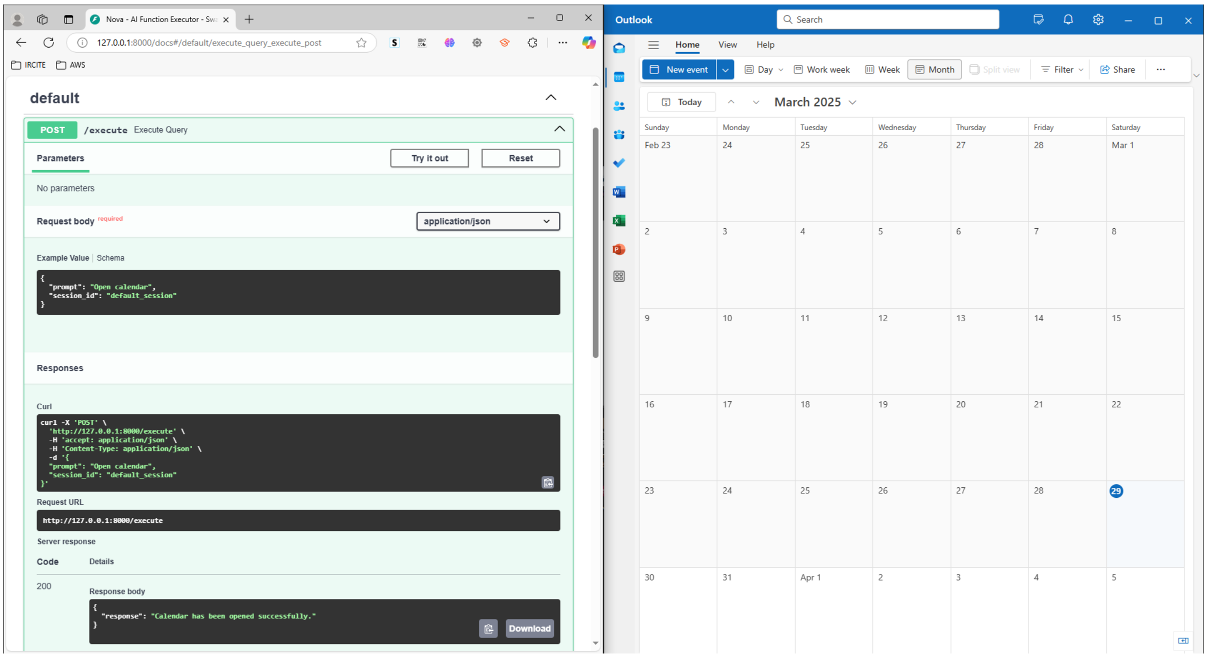
200

Response body

```
{
  "response": "Notepad has been opened successfully."
}
```

Download

- Open Calendar

- 

default

POST /execute Execute Query

Parameters — Try it out — Reset

No parameters

Request body required — application/json

Example Value | Schema

```
{
  "prompt": "Open calendar",
  "session_id": "default_session"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Open calendar",
  "session_id": "default_session"
}'
```

Request URL

```
http://127.0.0.1:8000/execute
```

Server response

Code Details

200

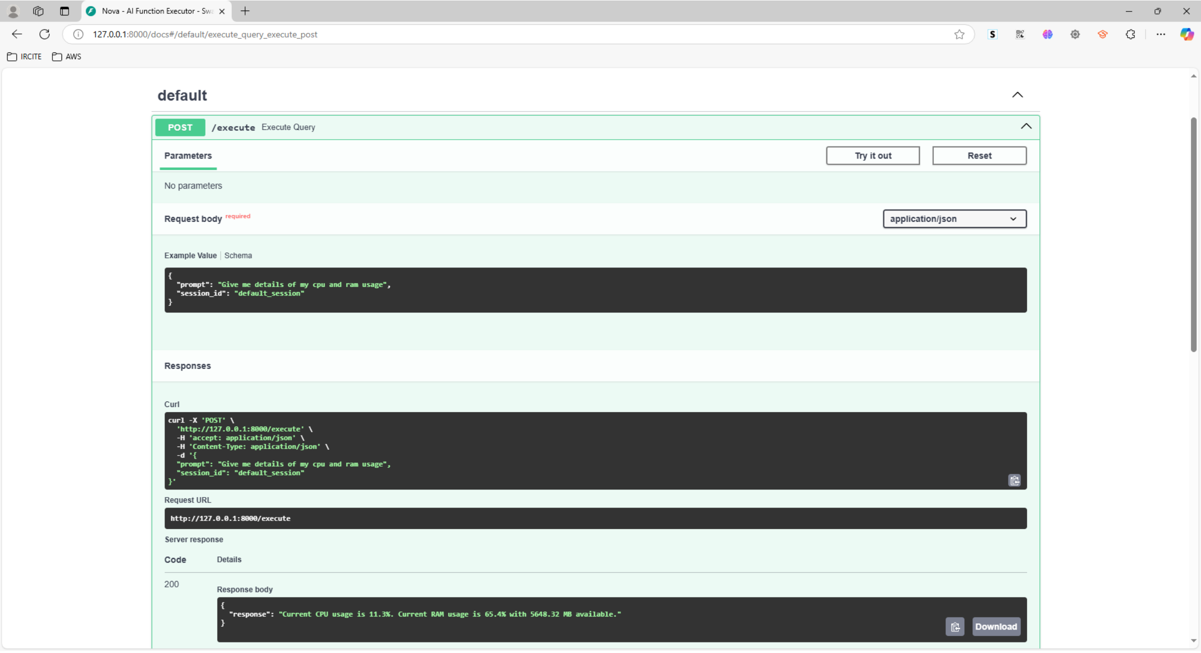
Response body

```
{
  "response": "Calendar has been opened successfully."
}
```

Download

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Feb 23	24	25	26	27	28	Mar 1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	Apr 1	2	3	4	5

- Retrieve CPU and RAM Usage Details

- 

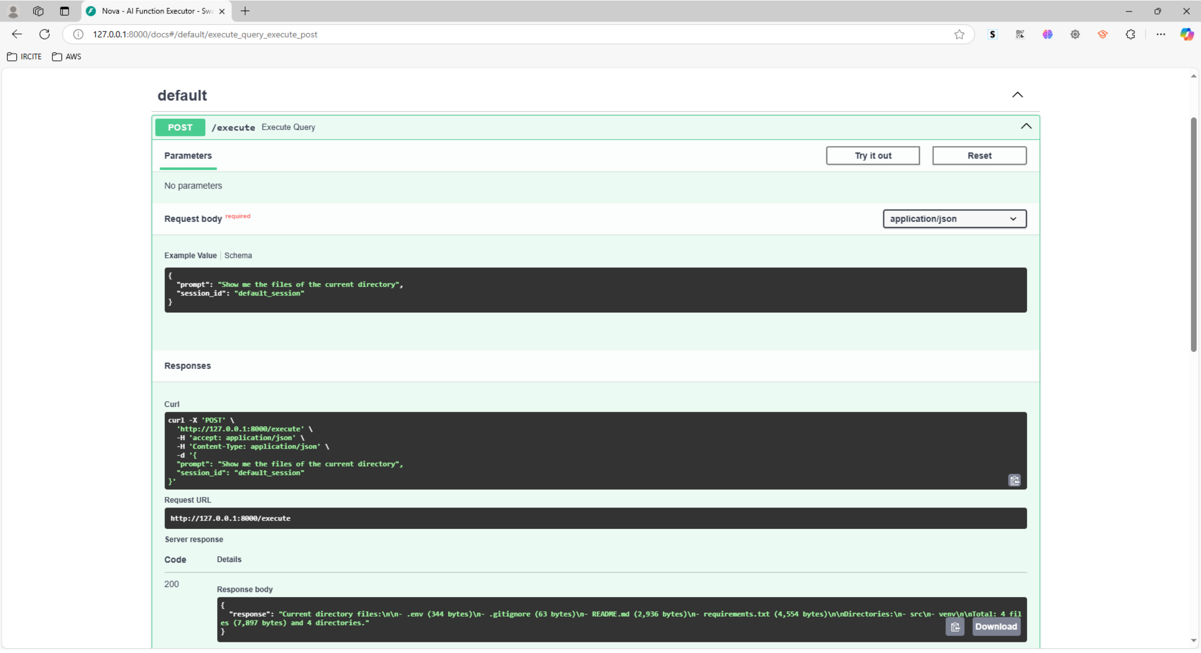
```
{
  "prompt": "Give me details of my cpu and ram usage",
  "session_id": "default_session"
}
```

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Give me details of my cpu and ram usage",
  "session_id": "default_session"
}'
```

```
http://127.0.0.1:8000/execute
```

```
{
  "response": "Current CPU usage is 11.3%. Current RAM usage is 65.4% with 5648.32 MB available."
}
```

- List Files in the Current Directory

- 

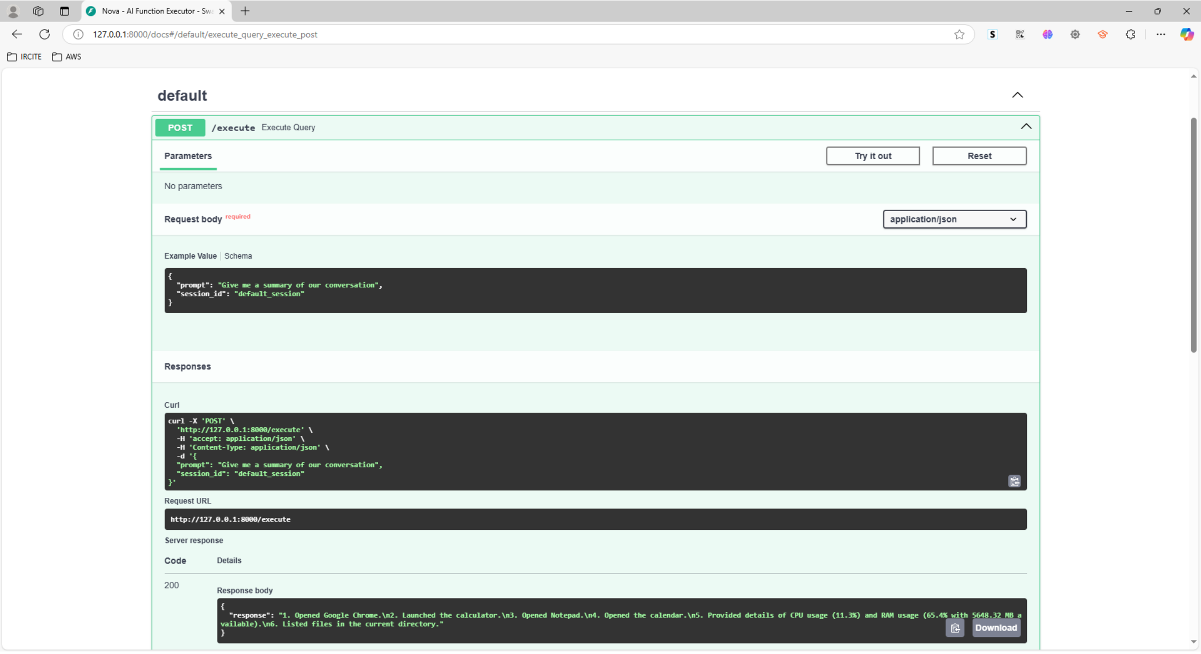
```
{
  "prompt": "Show me the files of the current directory",
  "session_id": "default_session"
}
```

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Show me the files of the current directory",
  "session_id": "default_session"
}'
```

```
http://127.0.0.1:8000/execute
```

```
{
  "response": "Current directory files:\n\n- .env (344 bytes)\n- .gitignore (63 bytes)\n- README.md (2,936 bytes)\n- requirements.txt (4,554 bytes)\n\nDirectories:\n- src\n- venv\n\nTotal: 4 fil
es (7,897 bytes) and 4 directories."
}
```

- Ask for Conversation Summary

- 

No parameters

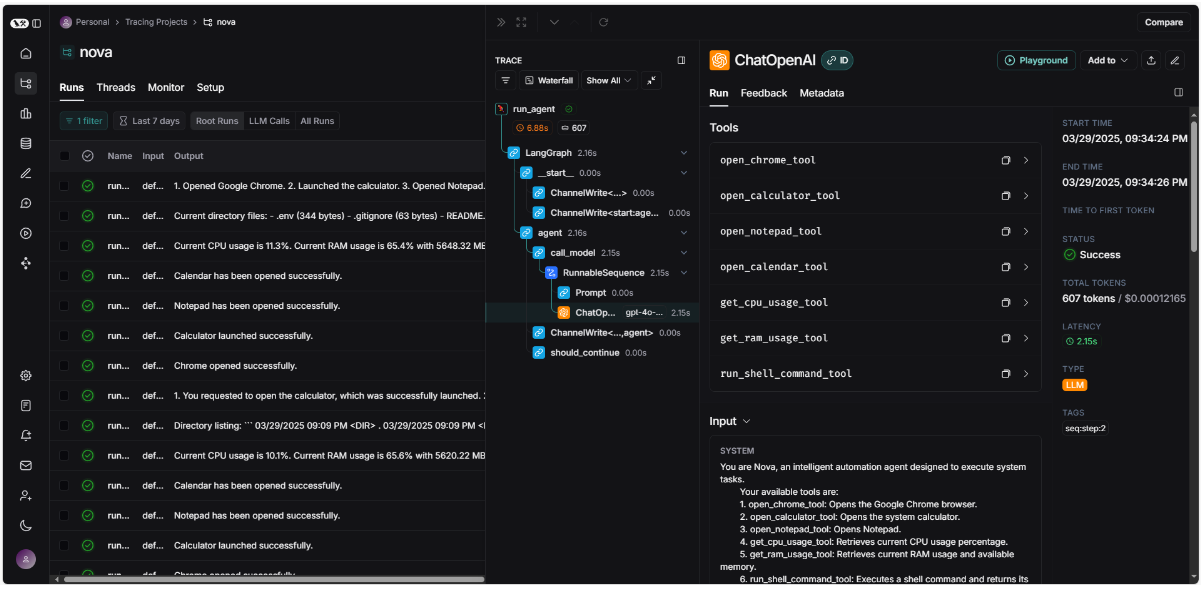
```
{
  "prompt": "Give me a summary of our conversation",
  "session_id": "default_session"
}
```

```
curl -X 'POST' \
  'http://127.0.0.1:8000/execute' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "prompt": "Give me a summary of our conversation",
  "session_id": "default_session"
}'
```

```
http://127.0.0.1:8000/execute
```

200

```
{
  "response": "1. Opened Google Chrome.\n2. Launched the calculator.\n3. Opened Notepad.\n4. Opened the calendar.\n5. Provided details of CPU usage (11.3%) and RAM usage (65.4% with 5648.32 MB a
vailable).\n6. Listed files in the current directory."
}
```

- **Langsmith Monitoring:**
  - LangSmith Logs
  - 

	Name	Input	Output
✓	run...	def...	1. Opened Google Chrome. 2. Launched the calculator. 3. Opened Notepad.
✓	run...	def...	Current directory files: - .env (344 bytes) - .gitignore (63 bytes) - README.
✓	run...	def...	Current CPU usage is 11.3%. Current RAM usage is 65.4% with 5648.32 MB
✓	run...	def...	Calendar has been opened successfully.
✓	run...	def...	Notepad has been opened successfully.
✓	run...	def...	Calculator launched successfully.
✓	run...	def...	Chrome opened successfully.
✓	run...	def...	1. You requested to open the calculator, which was successfully launched.
✓	run...	def...	Directory listing: ``` 03/29/2025 09:09 PM <DIR> . 03/29/2025 09:09 PM <
✓	run...	def...	Current CPU usage is 10.1%. Current RAM usage is 65.6% with 5620.22 MB
✓	run...	def...	Calendar has been opened successfully.
✓	run...	def...	Notepad has been opened successfully.
✓	run...	def...	Calculator launched successfully.
✓	run...	def...	Chrome opened successfully.

```
run_agent ✓
  6.88s  607
  LangGraph 2.16s
    __start__ 0.00s
      ChannelWrite<...> 0.00s
      ChannelWrite<start:age... 0.00s
    agent 2.16s
      call_model 2.15s
        RunnableSequence 2.15s
          Prompt 0.00s
          ChatOp... gpt-4o-... 2.15s
      ChannelWrite<...,agent> 0.00s
      should_continue 0.00s
```

Tools

- open_chrome_tool
- open_calculator_tool
- open_notepad_tool
- open_calendar_tool
- get_cpu_usage_tool
- get_ram_usage_tool
- run_shell_command_tool

START TIME
03/29/2025, 09:34:24 PM

END TIME
03/29/2025, 09:34:26 PM

TIME TO FIRST TOKEN

STATUS
✓ Success

TOTAL TOKENS
607 tokens / $0.00012165

LATENCY
2.15s

TYPE
LLM

TAGS
seq:step:2

Input

SYSTEM

You are Nova, an intelligent automation agent designed to execute system tasks.
    Your available tools are:
    1. open_chrome_tool: Opens the Google Chrome browser.
    2. open_calculator_tool: Opens the system calculator.
    3. open_notepad_tool: Opens Notepad.
    4. get_cpu_usage_tool: Retrieves current CPU usage percentage.
    5. get_ram_usage_tool: Retrieves current RAM usage and available memory.
    6. run_shell_command_tool: Executes a shell command and returns its