

Develop Generative AI solutions with Azure OpenAI Service

Thada Wangthammang

Technical Lead / Platform Engineer Team Lead
T. T. Software Solution

Microsoft AI Roadshow in Songkhla, Feb 26, 2025

Thada Wangthammang, (Mild)
Site Reliability Engineer and Team Lead.

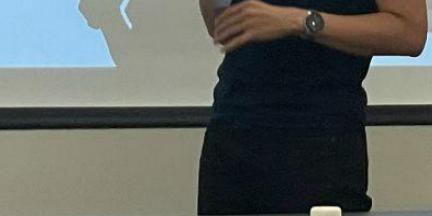
at:

T.T. Software Solution: tt-ss.net

WRM Software: wrmsoftware.com

Microsoft MVP, Area: Microsoft Azure
Cloud Native
Admin of ThaiType

THAI
TYPE



Thada Wangthammang, (Mild)
Tech Lead / Platform Engineer Lead

T.T. Software Solution: tt-ss.net



Microsoft MVP, Microsoft Azure, Cloud Native
Admin of ThaiType





T.T. SOFTWARE SOLUTION



We are Software Solution Expert made by MVP Technical Team

We are the leader of Microsoft Technology in Thailand,
developing business solutions with ASP.NET, Azure and C# by MVP team and experts.

Interested, please contact 086-899-6243 (Bangkok office) nakorn@tt-ss.net T. T. Software Solution @ttss

[Our Services](#)

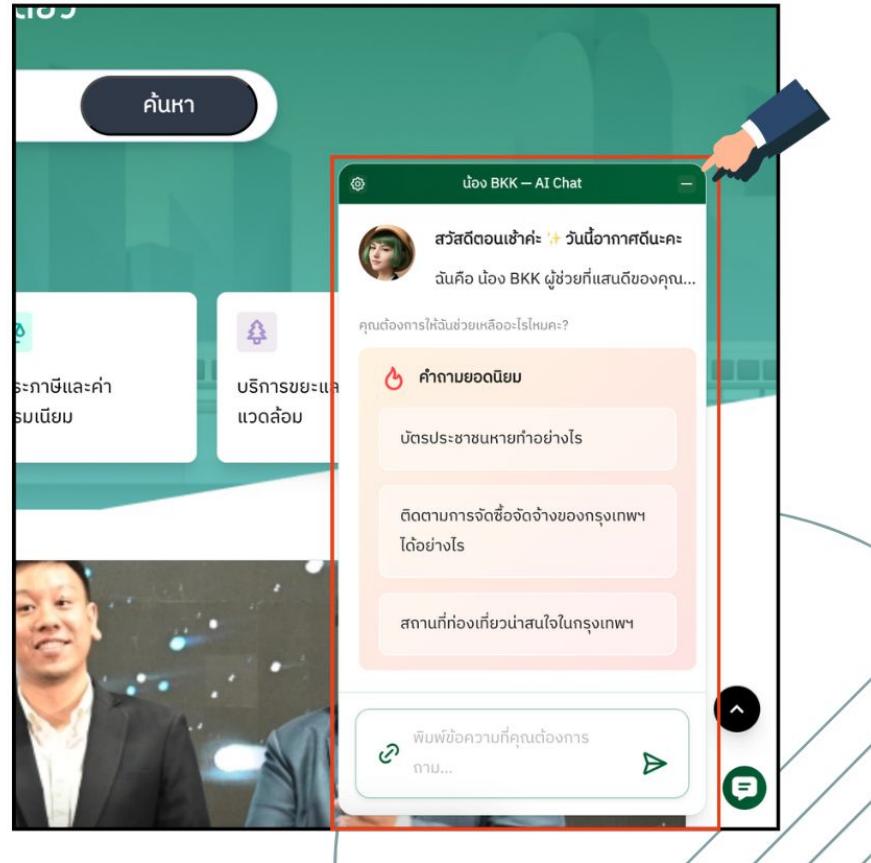
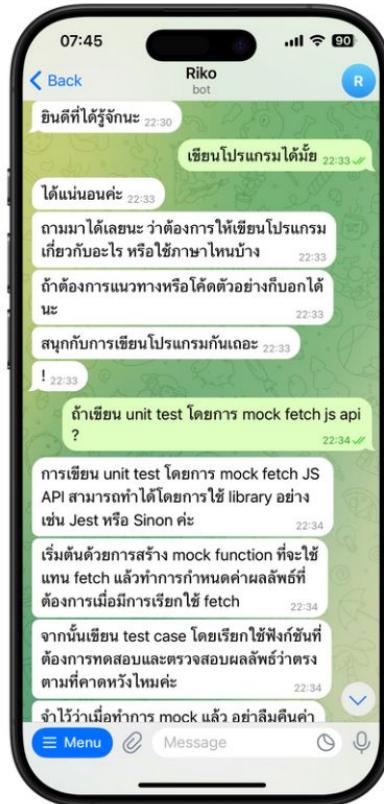
[Contact us](#)

Storm Manager (USA)

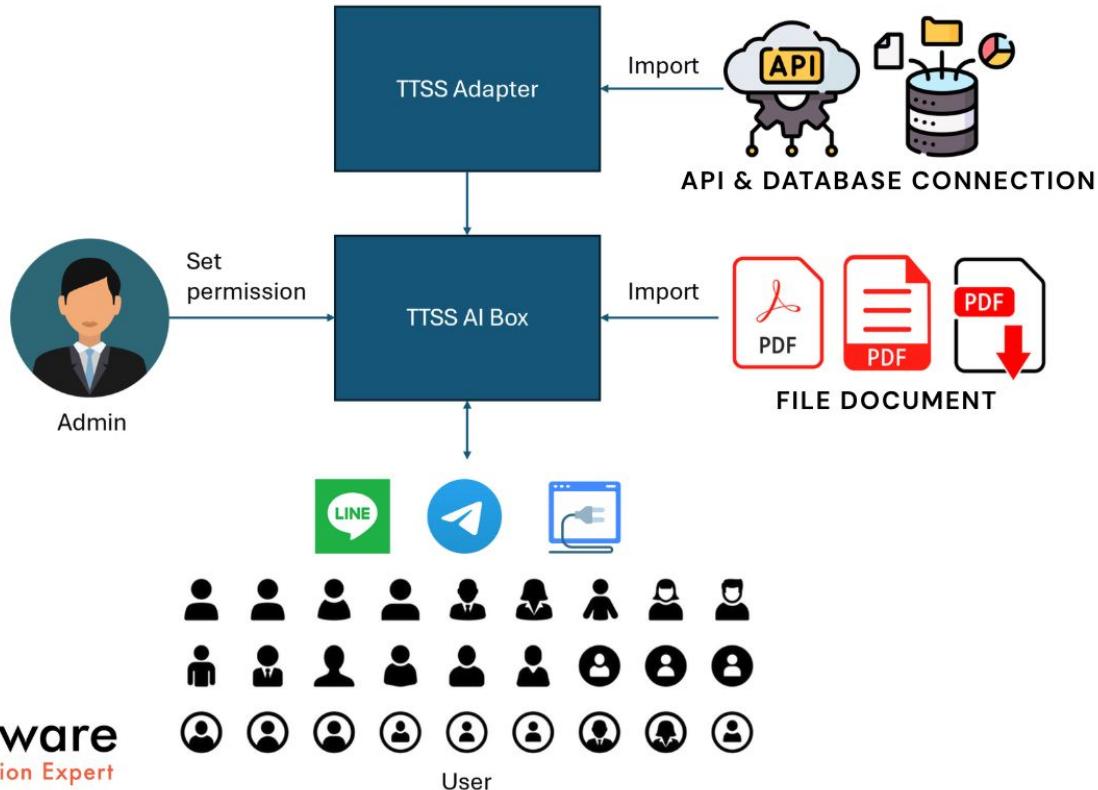
Storm Manager will now be the emergency response system for utilities in 9 states, covering about 35 million people



TT AI



CONCEPT



T.T. Software
Software Solution Expert



ไทยไทป์

4.3K likes · 6K followers



Posts

About

Mentions

Reels

Photos

Videos

More ▾



Agenda

- Introduction
- What we build?
- Telegram Bot with Grammy (TypeScript)
- Azure Open AI Services
- How does bot work?
- Demo



"Ever tried to remember
what you spent last week? Or lost a receipt?
Expense tracking is messy,
and most people just give up."

What If We Had an AI Assistant for Boring Tasks?



Imagine a world where AI handles tedious work for you...

- ◆ No more manually tracking expenses,

AI extracts & categorizes them automatically.



With AI, you focus on what matters, while it handles the rest!

Simply chat with the AI to log your expenses effortlessly

Text

68.00 THB to Tonhorm for a bill payment using ABC Bank

Image



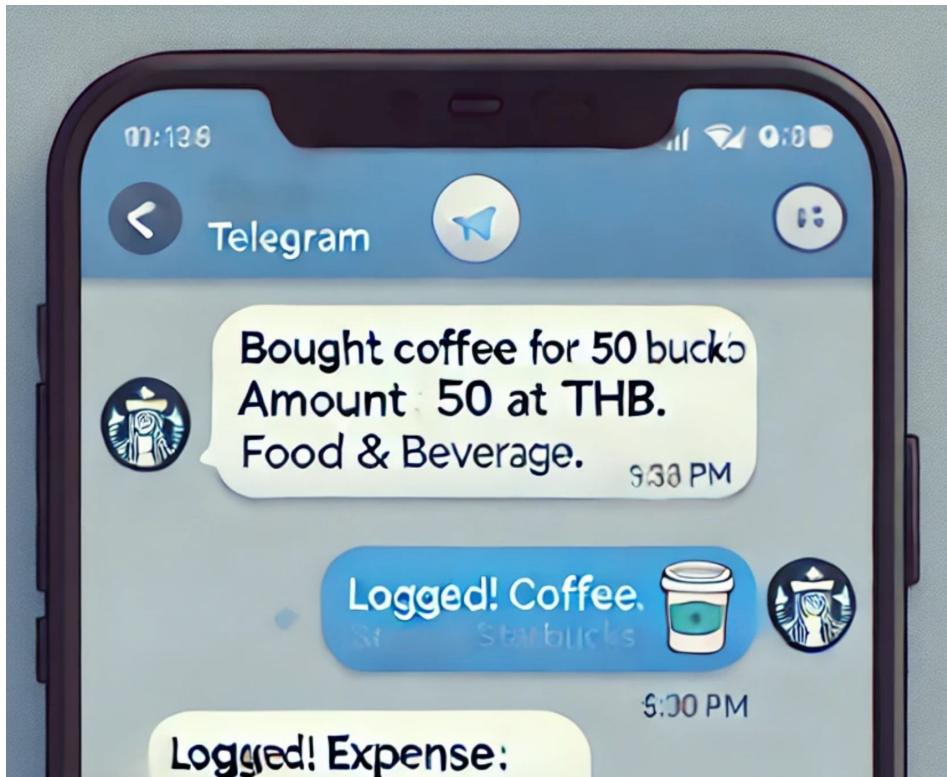
Building a Telegram Expense Tracker Bot with Azure OpenAI & TypeScript

(Making Expense Tracking Effortless with AI)

What We Build?

-  A Telegram AI Bot that:
 - Understands **messages and receipts** and extracts expense details.
 - Stores chat history in **Azure Storage Table**.
 - Extracts and Categorizes expenses using **Azure OpenAI**.
 - Saves structured data to **Notion Database**.
-  Goal: Automate expense tracking & minimize manual work.

User Journey – How It Works?



🚀 Steps:

- ① User sends a message on Telegram.
- ② Bot extracts key details using AI.
- ③ Data is stored & categorized.
- ④ Expense is saved in Notion for tracking.

Why Not Just Use ChatGPT? (Understanding AI API)



Key Difference:

ChatGPT remembers past chats – AI API does NOT!

We need to store previous messages manually.

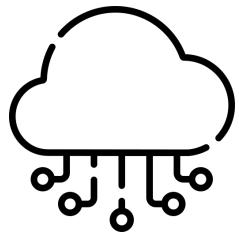


Solution:

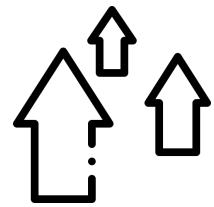
Store chat history in Azure Storage Table.

Design an efficient schema for fast retrieval.

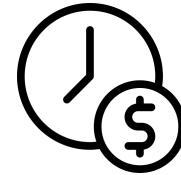
What's Serverless



Abstract of servers

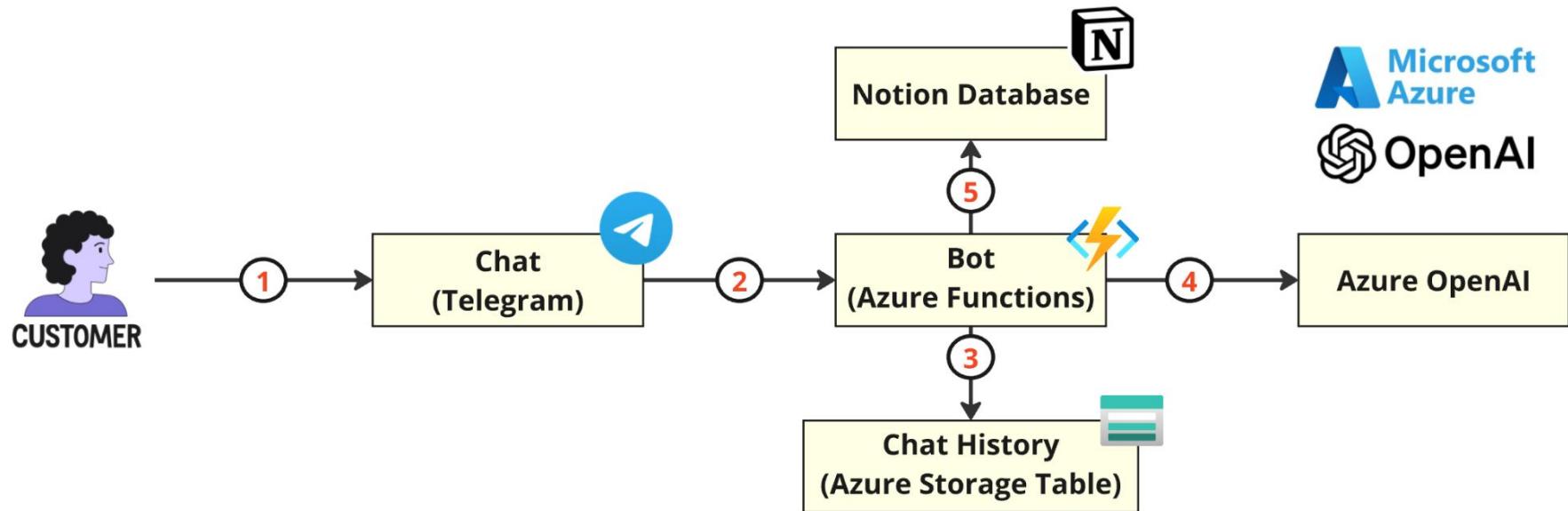


Event Driven /
Instant Scale

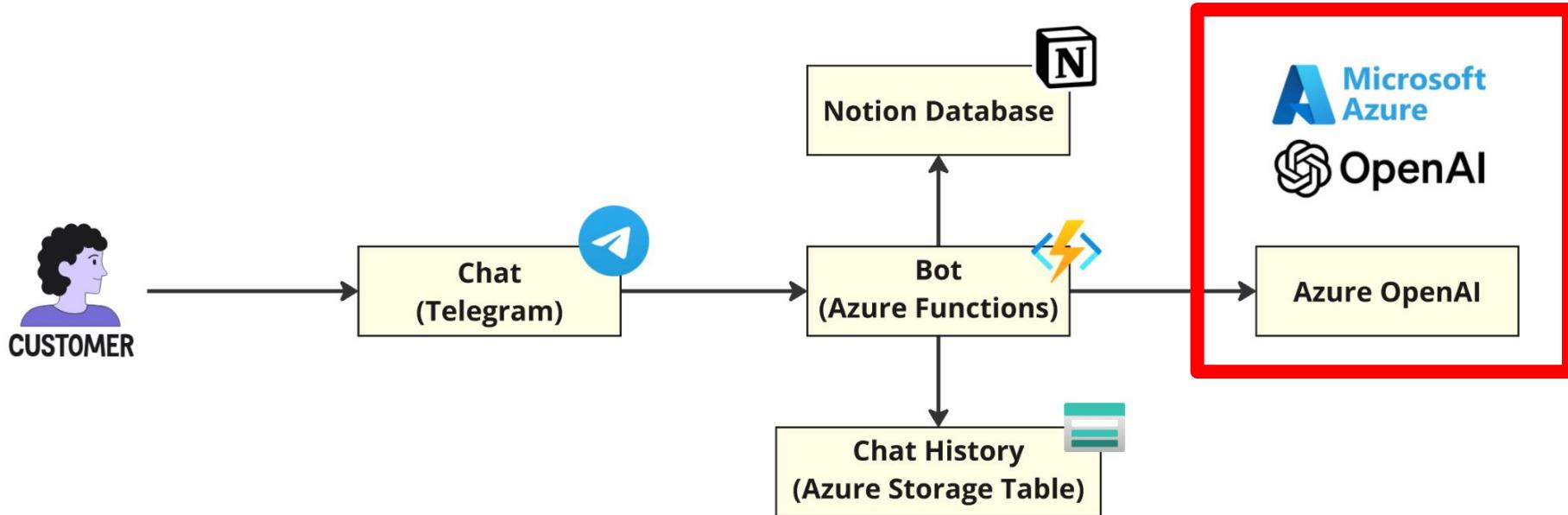


Pay as use

Architecture – How Everything Connects?



Architecture – How Everything Connects?



Extracting Expense Data (AI Processing)



AI Converts Text into Structured Data

Input: "Bought coffee for 50 THB at Starbucks."

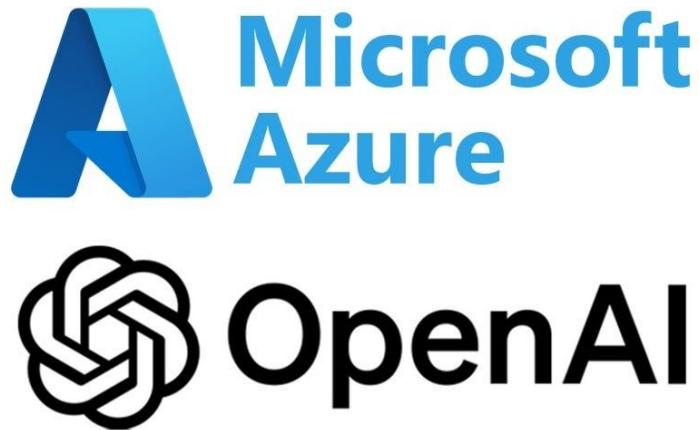


```
{  
  "amount": 50,  
  "category": "Food & Beverage",  
  "merchant": "Starbucks"  
}
```

✓ AI automatically categorizes & structures expenses.

What is Azure OpenAI Service?

- ◆ AI-Powered Cloud Service by Microsoft & OpenAI
- Azure OpenAI Service provides REST API access to OpenAI's powerful language models including **o3-mini**, **o1**, **o1-mini**, **GPT-4o**, **GPT-4o mini**, **GPT-4 Turbo** with **Vision**, **GPT-4**, **GPT-3.5-Turbo**, and **Embeddings** model series.



Azure OpenAI Service is

- - ◆ Enterprise-Grade AI with Built-in Security
 - Data privacy – prompts & responses are not used for model training.
 - ISO, SOC, HIPAA compliance for regulated industries
- - ◆ Scalable & Integrated with Azure Ecosystem
 - Works seamlessly with Azure Functions, Storage, Cosmos DB, and Logic Apps.
 - High availability & global scaling for production workloads.

How does Azure OpenAI Extract Text/Image?

- We use GPT 4o/GPT 4o-mini which is multi-model that can understand both text and images
- Using prompt engineering
- Azure OpenAI Structured Output

Using Prompt Engineering for Expense Extraction

-  Example Task: Extract structured data from user messages:
 - Extract: Memo, Amount, and Category.
 - Determine: **dateTimeUtc** based on conversation context.



Extract structured data from the following message:

- Identify the ****memo**** (description of the expense).
- Extract the ****amount**** spent.
- Categorize the expense (e.g., Food, Transport, Shopping).
- Determine the ****dateTimeUtc**** relative to the conversation timestamp.

Next Question after prompt, how we structure the data?

GPT-4o on ChatGPT

กินข้าว 80 บาทเมื่อวาน

Extract structured data from the following message:

- Identify the **memo** (description of the expense).
- Extract the **amount** spent.
- Categorize the expense (e.g., Food, Transport, Shopping).
- Determine the **dateTimeUtc** relative to the conversation timestamp.

Here's the extracted structured data from the message:

json

 Copy

```
{  
  "memo": "กินข้าว",  
  "amount": 80,  
  "category": "Food",  
  "dateTimeUtc": "2025-02-25T00:00:00Z"  
}
```

Azure OpenAI Structured Output

- And yes, Azure OpenAI support structured Output
- both Azure OpenAI and OpenAI (ChatGPT) can be use the same library without changing the code

Structured outputs

Article • 02/11/2025 • 5 contributors

Feedback

Choose a language or API

C# Python REST

In this article

[Supported models](#)

[API support](#)

[Getting started](#)

[Function calling with structured outputs](#)

[Supported schemas and limitations](#)

Structured outputs make a model follow a [JSON Schema[↗]](#) definition that you provide as part of your inference API call. This is in contrast to the older [JSON mode](#) feature, which guaranteed valid JSON would be generated, but was unable to ensure strict adherence to the supplied schema. Structured outputs are recommended for function calling, extracting structured data, and building complex multi-step workflows.

TypeScript Structured Output by OpenAI Library

```
1 import OpenAI from "openai";
2 import { zodResponseFormat } from "openai/helpers/zod";
3 import { z } from "zod";
4
5 const openai = new OpenAI();
6
7 const CalendarEvent = z.object({
8   name: z.string(),
9   date: z.string(),
10  participants: z.array(z.string()),
11 });
12
13 const completion = await openai.beta.chat.completions.parse({
14   model: "gpt-4o-2024-08-06",
15   messages: [
16     { role: "system", content: "Extract the event information." },
17     { role: "user", content: "Alice and Bob are going to a science fair on Friday." },
18   ],
19   response_format: zodResponseFormat(CalendarEvent, "event"),
20 });
21
22 const event = completion.choices[0].message.parsed;
```

<https://platform.openai.com/docs/guides/structured-outputs>

TypeScript Structured Output by OpenAI Library

```
13 const completion = await openai.beta.chat.completions.parse({
14   model: "gpt-4o-2024-08-06",
15   messages: [
16     { role: "system", content: "Extract the event information." },
17     { role: "user", content: "Alice and Bob are going to a science fair on Friday." },
18   ],
19   response_format: zodResponseFormat(CalendarEvent, "event"),
20 });
21
```

<https://platform.openai.com/docs/guides/structured-outputs>

Using Prompt Engineering for Expense Extraction (Recap)

-  Example Task: Extract structured data from user messages:
 - Extract: Memo, Amount, and Category.
 - Determine: **dateTimeUtc** based on conversation context.



Extract structured data from the following message:

- Identify the ****memo**** (description of the expense).
- Extract the ****amount**** spent.
- Categorize the expense (e.g., Food, Transport, Shopping).
- Determine the ****dateTimeUtc**** relative to the conversation timestamp.

Expense Structured Output

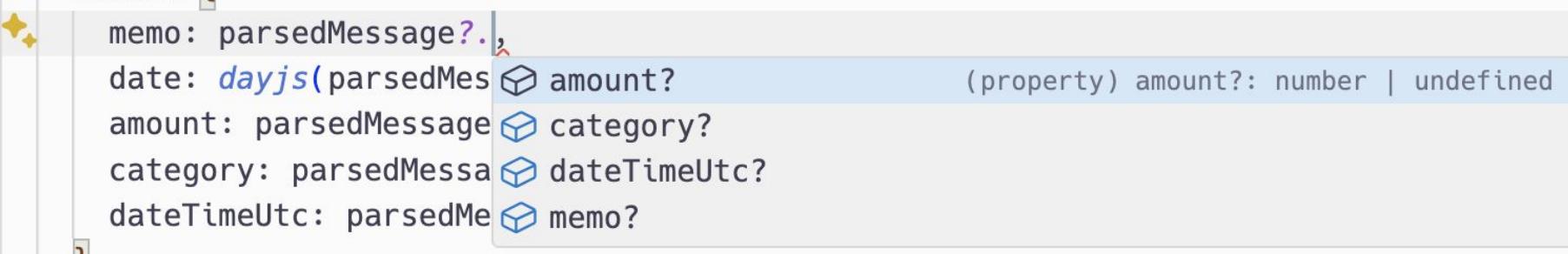
- OpenAI Library Support Zod Schema for Structured Output

```
export const expenseAgentResponseSchema = z.object({
    dateTimeUtc: z.string().optional(),
    amount: z.number().optional(),
    category: z.string().optional(),
    memo: z.string().optional(),
})
```

With Zod Schema Adapter, type-safety

- OpenAI Library Support Zod Schema for Structured Output

```
return {  
  memo: parsedMessage?.  
  date: dayjs(parsedMes  
  amount: parsedMessage.  
  category: parsedMessa  
  dateT  
  dateT  
}  
}
```

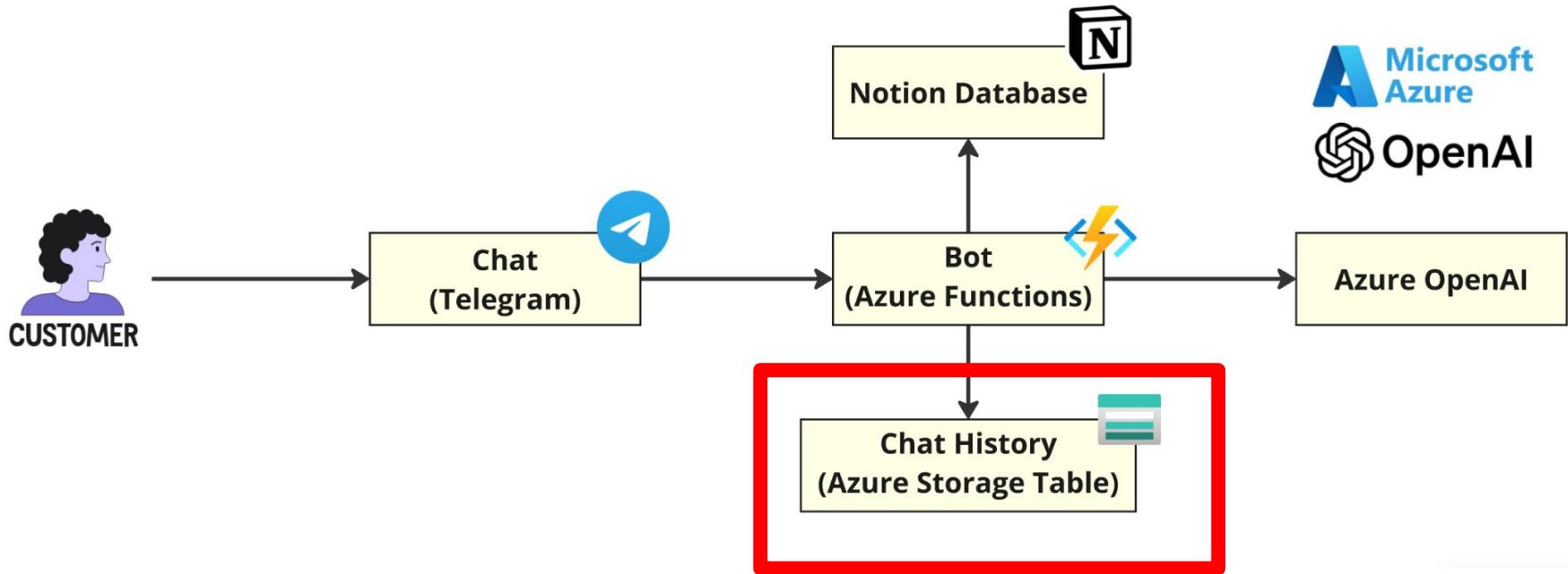


All Prompts used in single API call.

- Tip: OpenAI API don't know the datetime. So, we need to send the current date, that's it!

```
const chatCompletion = await this.client.beta.chat.completions.parse({
  messages: [
    ...SystemRole[character],
    ...CharacterRole[this.characterRole],
    ...this.generateSystemMessages(`Current Date (UTC): ${dayjs().toISOString()}`)],
    ...this.generatePreviousMessages(previousMessages),
    ...this.generateTextMessages(messages),
  ],
  model: this.model,
  response_format: zodResponseFormat(expenseAgentResponseSchema, "expense"),
});
```

Architecture – How Everything Connects?



How We Store Chat History?

- 📌 **Challenge:** AI API doesn't remember past messages, requiring a storage mechanism for message history.
- 📌 **Solution:** Use Azure Storage Table (NoSQL) with an optimized schema for fast retrieval and efficient sorting.

What is Azure Storage Table

 A Cost-Effective, Scalable NoSQL Database

 Cheap & Serverless

- Pay-as-you-go pricing – only charged for stored data and operations.
- No need to manage infrastructure or database servers.

 Scalable & High-Performance

- Handles millions of records efficiently.
- Partitioned storage enables fast lookups based on key design.

What is Azure Storage Table (2)

Some Learning Curve

- Because Azure Storage Table cannot be sorted, it requires schema optimization (PartitionKey, RowKey).
- NoSQL querying – must design for specific access patterns.

Seamless Upgrade to Azure Cosmos Table

- Same API, same codebase – just change the backend.
- Low-latency global distribution for extreme workloads.

Message History

Timestamp	Message
00000001000	Hello World 1
00000002000	Hello World 2
00000003000	Hello World 3
00000004000	Hello World 4

ASC Sort



Incoming Message

Timestamp	Message
00000001000	Hello World 1
00000002000	Hello World 2
00000003000	Hello World 3
00000004000	Hello World 4
00000005000	Hello World 4

Incoming Message



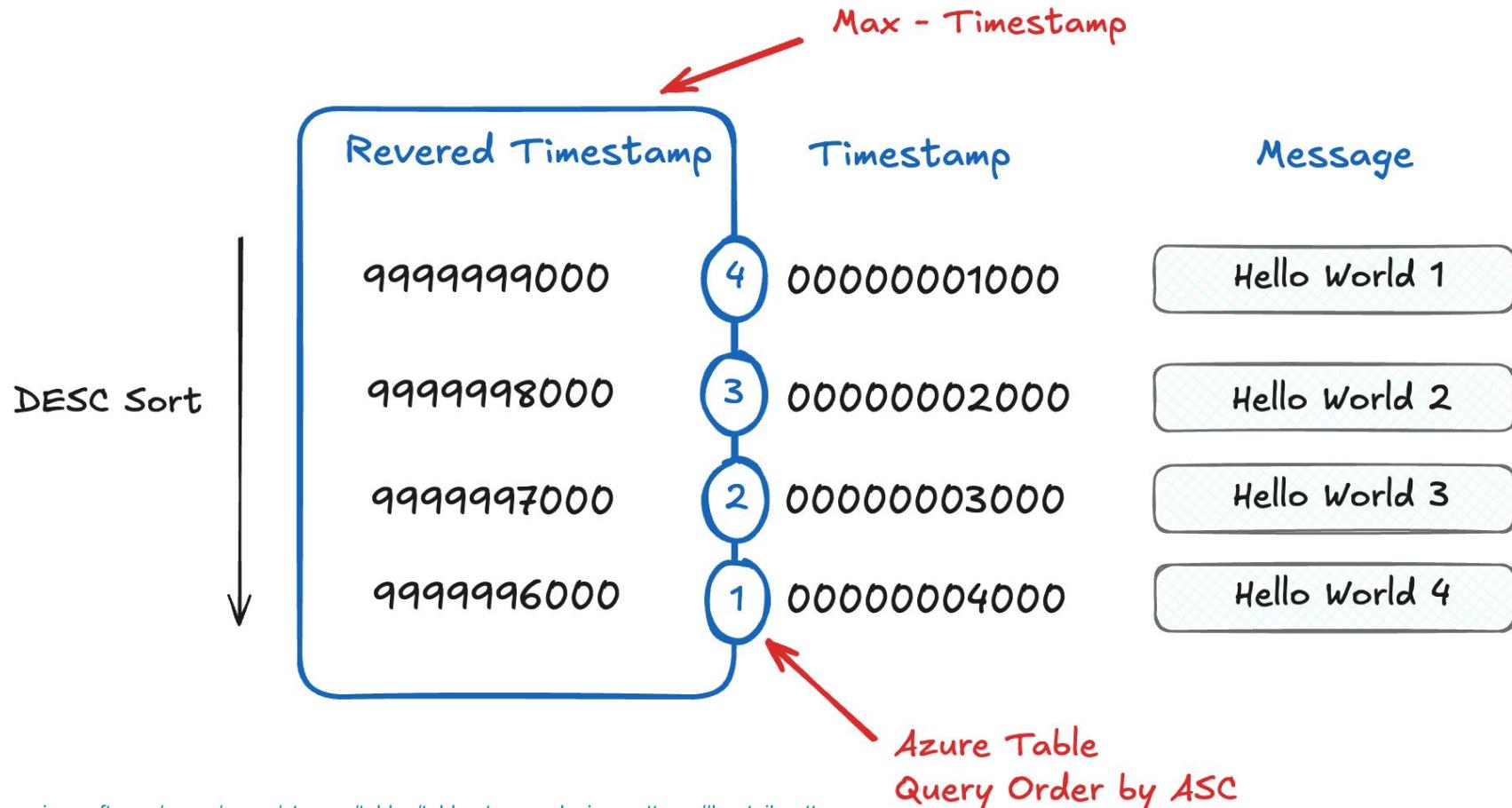
How to retrieve messages history without sorting?

Timestamp	Message
00000001000	Hello World 1
00000002000	Hello World 2
00000003000	Hello World 3
00000004000	Hello World 4
00000005000	Hello World 4

A blue arrow points from the text "Messages History" to the fourth message row (timestamp 00000004000). The entire row for timestamp 00000004000 is highlighted with a blue border, indicating it is the current message being retrieved.

Messages History

Solution: Log Tail Pattern



Azure Storage Table Schema Design

- ◆ Schema Design for High-Performance Retrieval:

- PartitionKey: **{YYYY}-{UserId:20}**
 - Groups messages by year and user ID (padded to 20 characters) for efficient queries.
 - Enables partition-level access for scalability.
- RowKey: **{LogTailTimestamp}-{MessageHash:10}**
 - Descending timestamp for log-tail pattern, ensuring recent messages are retrieved first.
 - SHA-3 hash (first 10 characters) of the message ensures uniqueness while reducing storage footprint.

Example of PartitionKey & RowKey

{LogTailTimestamp}

PartitionKey	RowKey
2025-0000000005281769912	098259788981-2baa768d8b
2025-0000000005281769912	098259788983-5caf3fc11d
2025-0000000005281769912	098259789029-81ff22a65a

Example Data

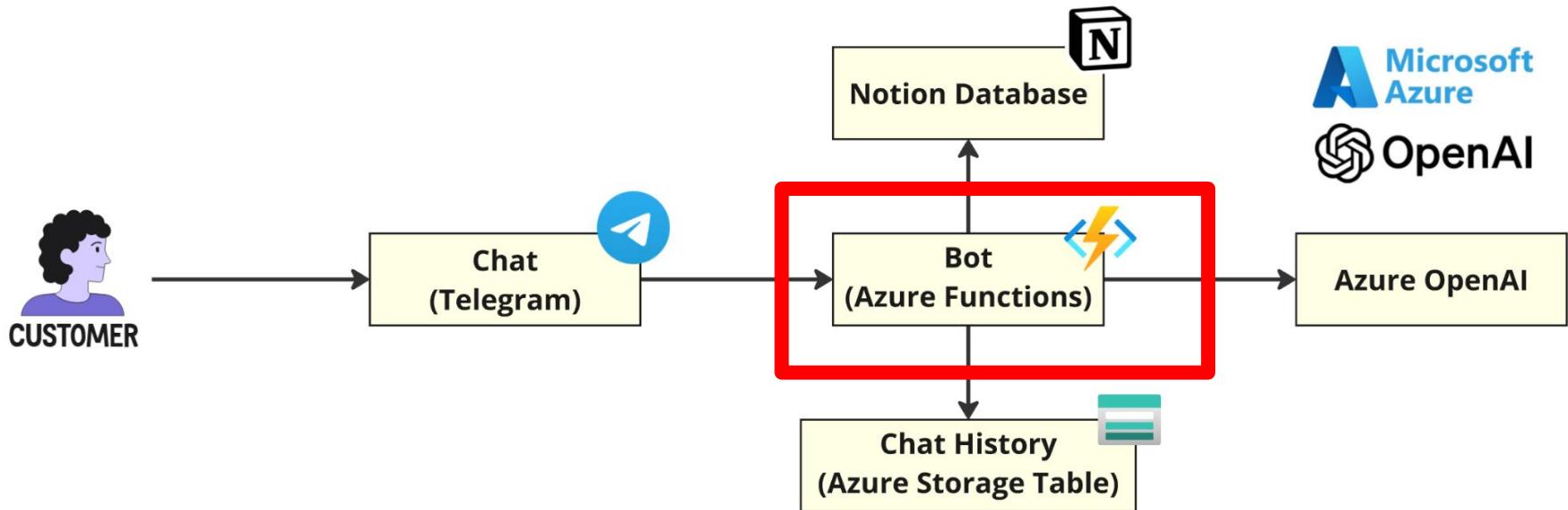
Microsoft Azure Sto

The screenshot shows the Microsoft Azure Storage Explorer interface. At the top, there's a toolbar with icons for Query, Import, Export, Add, Edit, Select All, Column Options, Delete, Table Statistics, and Refresh. The title bar says "MyBotBot". On the left, there are navigation icons for Home, Tables, Blobs, Queues, and Functions.

PartitionKey	RowKey	Timestamp	payload
2025-000000000008071676474	098259792916-7cd0809487	2025-02-22T06:51:24.9194657Z	อ้อ เข้าใจแล้วค่ะ พี่มายคือคนที่สร้างริโกะชื่นมา รู้สึกตื่นมากเลยนะที่มี
2025-000000000008071676474	098259792920-2f8cfa1594	2025-02-22T06:51:20.7029738Z	พี่มายคือคนที่สร้างເອົ້ນມາຍັງໄລ່ຮີໂກ
2025-000000000008071676474	098259792954-7bab6f14c7	2025-02-22T06:50:46.1575289Z	ໄມ້ຮັບພໍມາຍັດຕະ ແຕ່ມັນດູ້ເໝືອຈະປັບປຸງໃຫຍ່ ໂຮງກວນເລົາ
2025-000000000008071676474	098259792958-a8e3ee3ea9	2025-02-22T06:50:42.4657269Z	ຮັບພໍມາຍັນ
2025-000000000008071676474	098259792992-6a4e630ea5	2025-02-22T06:50:08.8647416Z	ຮີໂກໄມ່ແນ່ໃຈຕະ ແຕ່ດັ່ງນີ້ເຮືອງຂາກພູດຄຸຍເຖິງກັບຄວບຄົວຫຼືເຮືອງອື່ນ
2025-000000000008071676474	098259792995-04130da614	2025-02-22T06:50:05.2299077Z	ໄຄເປັນພ້ອຂອງຮີໂກ
2025-000000000008071676474	098259793483-33700f8115	2025-02-22T06:41:57.841863Z	ບັນທຶກຄໍາ ໃໃຈໜ້າ: Note ກາแฟໜົມ, 405 ນາທ ປະເທດ: Cafe ວັນທີ: Februa
2025-000000000008071676474	098259793486-4a3a9bc09c	2025-02-22T06:41:54.4108015Z	ມັນດູ້ຕ່າກພົມຕະຮີໂກ
2025-000000000006562752036	098259793495-80502d50ca	2025-02-22T06:41:45.0480836Z	ຕອນນີ້ເວລາ 06:41 ນ. ດະ ມີອາໄຫຫ້ວຍໄດ້ໄໝ? 😊
2025-000000000006562752036	098259793497-9229e0a498	2025-02-22T06:41:43.3680324Z	ຕອນນີ້ໂມງ
2025-000000000008071676474	098259793618-c16dac719f	2025-02-22T06:39:42.0544763Z	ບັນທຶກຄໍາ ໃໃຈໜ້າ: Note ກາແພິມ, 405 ນາທ ປະເທດ: Cafe ວັນທີ: Februa

Showing 1 to 100 of 204 cached items

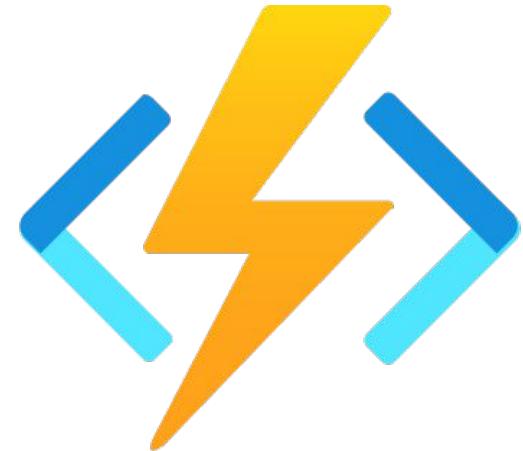
Architecture – How Everything Connects?



What are Azure Functions

- ◆ Serverless Compute for Event-Driven Applications
 - Run code on demand without managing infrastructure.
 - Auto-scales based on workload, reducing costs.

- ◆ Cost-Efficient & Scalable
 - Pay-per-execution model – only pay when the function runs.
 - Efficient for microservices, automation, and AI workflows.



What are Azure Functions

Execute your code in response to any event



```
// @azure/functions@3.5.x
import { Context, HttpRequest } from '@azure/functions';

export default function myFunction(context: Context, req: HttpRequest) {
    context.log('JavaScript HTTP trigger function processed a request.');

    // Perform work

    return {
        body: 'OK',
    };
}
```

GrammY: Telegram Bot Framework

The screenshot shows the homepage of the grammY.dev website. At the top, there's a dark header bar with browser controls (back, forward, search, etc.) and a URL field showing "grammy.dev". Below the header is a navigation bar with links: Guide, Learn, Plugins, Examples, Resources, Hosting, API Reference, and a language selector. To the right of the navigation are several icons for search, refresh, and user profile.

The main content area features a large logo "grammY" in blue and black, followed by the text "The Telegram Bot Framework." in bold black. Below this, a tagline "... have fun making bots." is displayed in a smaller italicized font. Two buttons are present: a teal "Get Started" button and a grey "Documentation" button.

On the right side of the page is a large, stylized blue "Y" graphic.

Below the main title, there are three callout boxes:

- Easy-to-use**: Shows a small emoji of a person at a beach. Description: "grammY makes creating Telegram bots so simple you already know how to do it."
- Flexible**: Shows a small emoji of a painter's palette. Description: "grammY is open and can be extended by plugins to make it fit exactly your needs."
- Scalable**: Shows a small emoji of a rocket ship. Description: "grammY has you covered when your bot gets popular and the traffic increases."

GrammY will support Azure Functions soon

github.com/grammyjs/grammY/pull/738

grammyjs / grammY

Type ⌘ to search

Code Issues 37 Pull requests 4 Actions Projects Security Insights

feat: add support for Azure Functions v4 #738

Open KnorpelSenf wants to merge 9 commits into `main` from `support-azure-v4`

Conversation 21 Commits 9 Checks 6 Files changed 1 +30 -14

KnorpelSenf commented on Jan 15

Adds a new webhook adapter following <https://learn.microsoft.com/en-us/azure/azure-functions/functions-node-upgrade-v4>

Fixes #737

feat: add support for Azure Functions v4 ... Verified ✓ f5e6e85

Reviewers

mildronize ✓

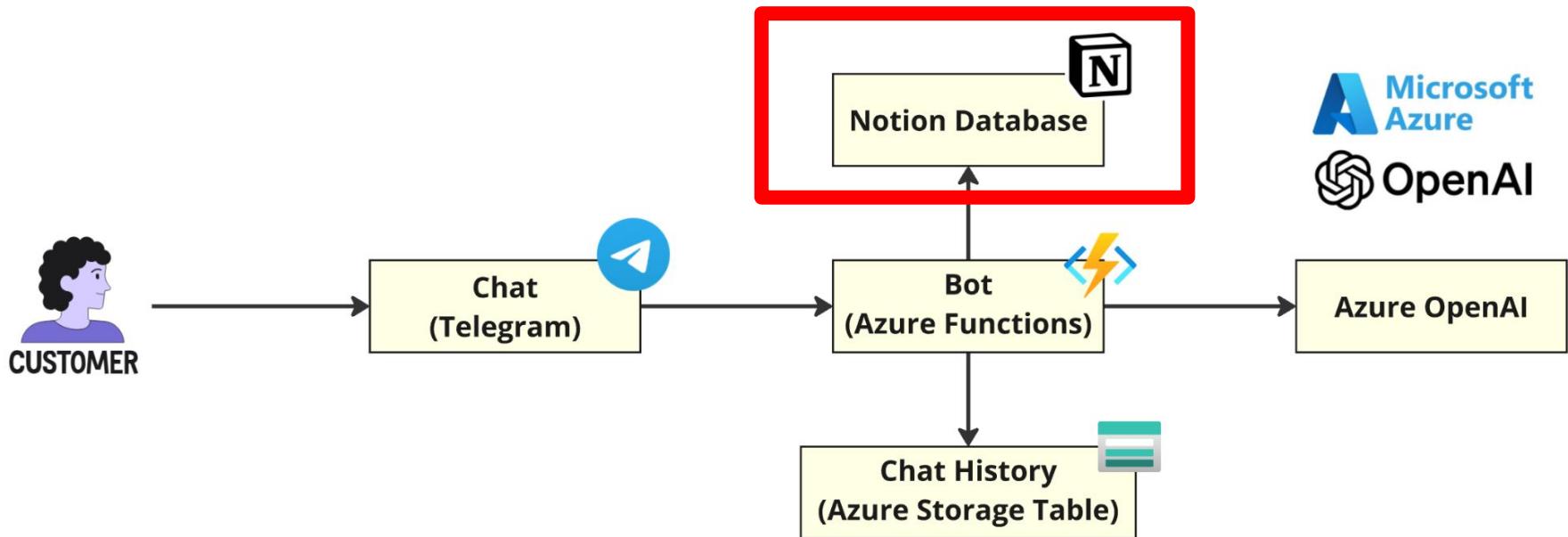
At least 1 approving review is required to merge this pull request.

Still in progress? Learn about draft PRs ⓘ

Assignees

No one assigned

Architecture – How Everything Connects?



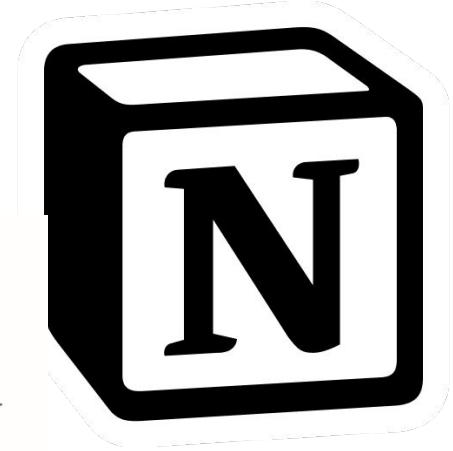
Notion API



```
curl https://api.notion.com/v1/users  
-H "Authorization: Bearer secret_token"  
  
{  
  "name": "Ada Lovelace",  
  "type": "person",  
  "person": {  
    "email": "ada@example.org",  
  },  
  ...  
}
```

Building blocks for developers

Aggregate data from many sources into your team's workspace. Spend less time context switching, and increase visibility across the software and services you rely on.



API reference

A technical guide to querying, retrieving, and updating pages, databases, blocks, users and more.

[Jump to the reference →](#)

Examples

A starting point to see what's possible with the API, so you can start bringing your own vision to life.

[Learn more →](#)

Partner with us

Built a public Notion integration? Become a technology partner to grow its adoption.

[Learn more →](#)

Notion Database

☰ Public Page / Demo Expense Tracker (Finbot)

Demo Expense Tracker (Finbot)

Table +

Aa Memo	Date	Category	# Amount	+	...
กาแฟ	February 24, 2025 7:01 AM	Food	30		
กาแฟ	February 24, 2025 7:01 AM	Food	20		
กาแฟ	February 23, 2025 10:10 PM	Food	50		
ข้าว	February 23, 2025 10:09 PM	Food	70		

+ New page

Try it by yourself

08:17 TELEGRAM 92

Public Page / Demo Expense Tracker (Finbot) Share ⭐ ...

Back Finbot F

อาหารสด 18:29

บันทึกค่าใช้จ่าย: Note Purchase at Big C Supercenter Hatyai 2, 821 บาท ประเภท: Groceries วันที่: February 23, 2025 18:28 18:29

แก้ไขรายการการเป็น อาหารสด 18:31

บันทึกค่าใช้จ่าย: Note Purchase at Big C Supercenter Hatyai 2, 821 บาท ประเภท: อาหารสด วันที่: February 23, 2025 11:28 18:31

แก้ไข note อาหารสด, ประเภท prep meal 18:32

บันทึกค่าใช้จ่าย: Note Purchase at Big C Supercenter Hatyai 2, 821 บาท ประเภท: prep meal วันที่: February 23, 2025 11:28 18:32

กาแฟ 80 21:51

บันทึกค่าใช้จ่าย: Note กาแฟ, 80 บาท ประเภท: Food วันที่: February 23, 2025 21:00 21:52

Menu Message

Demo Expense Tracker (Finbot)

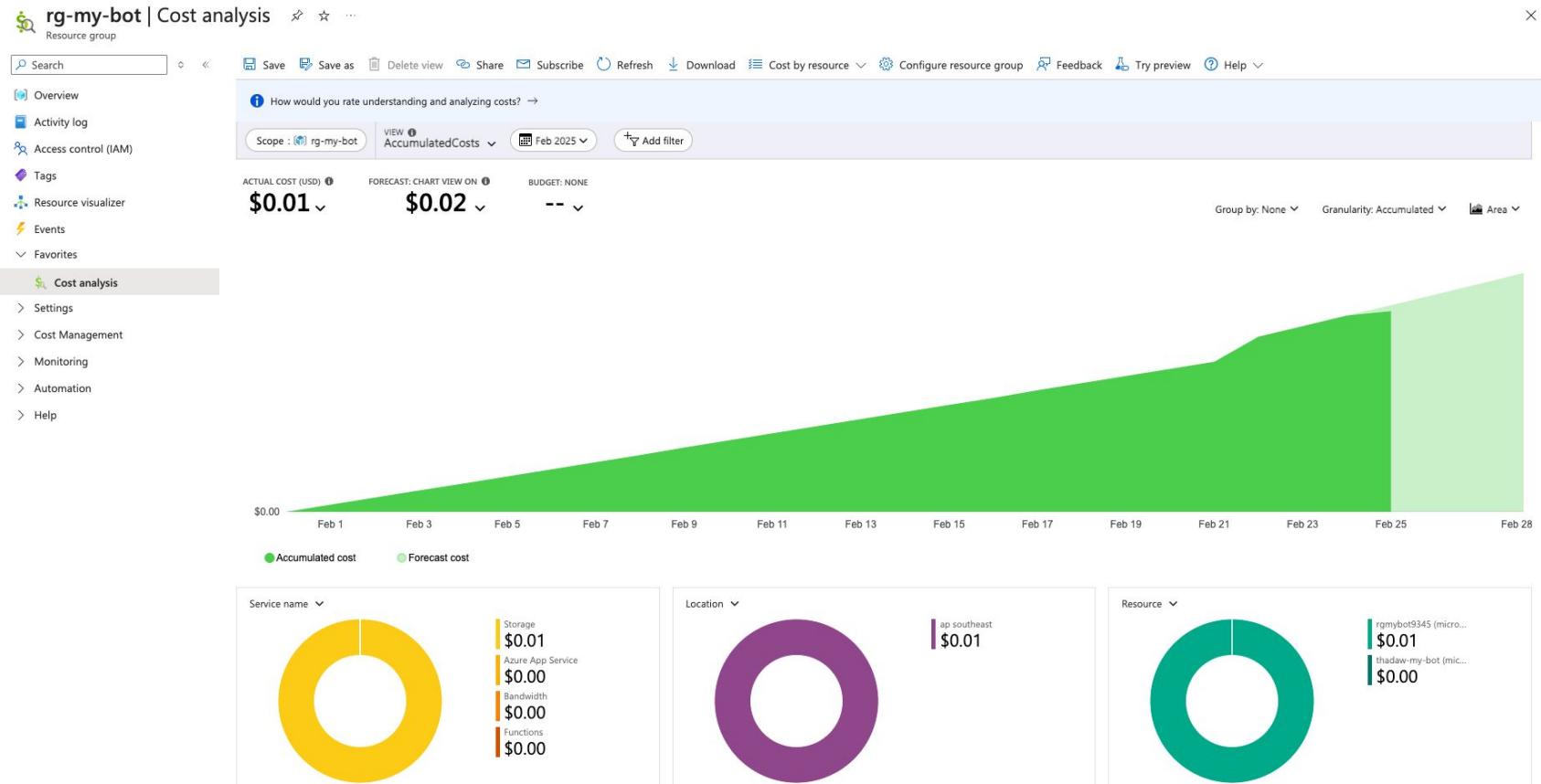
Table + ... + ▾

Aa Memo	Date	Category	# Amount
กาแฟ	February 24, 2025 7:01 AM	Food	30
กาแฟ	February 24, 2025 7:01 AM	Food	20
กาแฟ	February 23, 2025 10:10 PM	Food	50
ช้าว	February 23, 2025 10:09 PM	Food	70

+ New page



Cost?



Azure Storage Table

Storage Accounts

Region:	Type:	Tier:	Redundancy:
Southeast Asia	Table Storage	Standard	LRS

Capacity

1 GB × \$0.0450 = \$0.05
Per GB

Storage transactions

100 Transaction units (10,000 transactions) × \$0.0004 = \$0.04
Per unit

Upfront cost	\$0.00
Monthly cost	\$0.08

Azure Functions

[Azure Functions](#)



Consumption tier, Pay as you go, 128 MB memory, ...



Upfront: \$0.00

Monthly: \$0.20

Azure Functions

Region:

Tier:

Southeast Asia

Consumption

The first 400,000 GB/s of execution and 1,000,000 executions are free.

Executions

Memory size:

128



200



2000000



\$0.00

Execution time (in milliseconds)

Executions per month

Requests

2,000,000

Execution count



\$0.20

Upfront cost

\$0.00

github.com/mildronize/lumebot

mildronize / lumebot

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

lumebot Public

Your main branch isn't protected

Protect this branch

Unpin Unwatch Fork Star

main 6 Branches 9 Tags Go to file Add file Code

About All in one bot on Telegram Readme Activity

github.com/mildronize/finbot

mildronize / finbot

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

finbot Public

main 1 Branch 0 Tags Go to file Add file Code

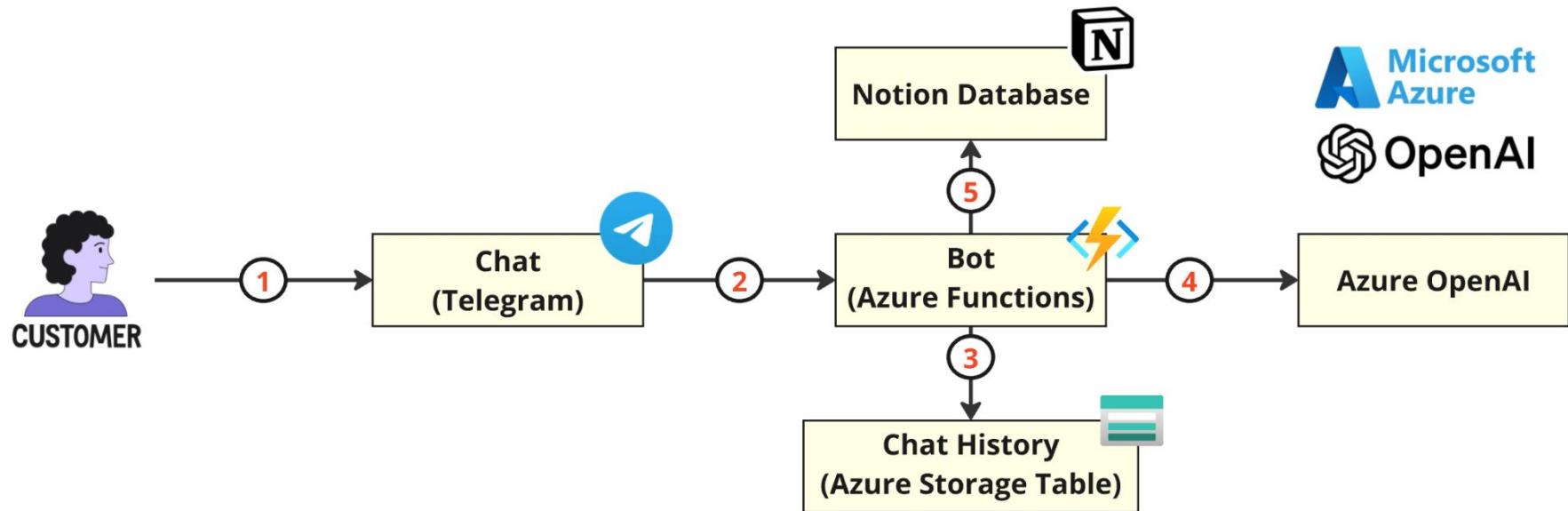
mildronize feat: migrate to Azure OpenAI 8e716c3 · 2 days ago 72 Commits

.vscode format: format it last month

scripts chore: remove comment last month

About Expense Bot (based on lumebot) Readme Activity 0 stars 1 watching

Architecture – How Everything Connects?



Contact us

We are ready to serve our customers and partners in the best possible way.

Technical Department

📞 08 6899 6243

✉️ nakorn@tt-ss.net

Customer Department

📞 09 0161 9166

✉️ watcharapong@tt-ss.net

HR Department

📞 06 1018 1275

✉️ hr@tt-ss.net

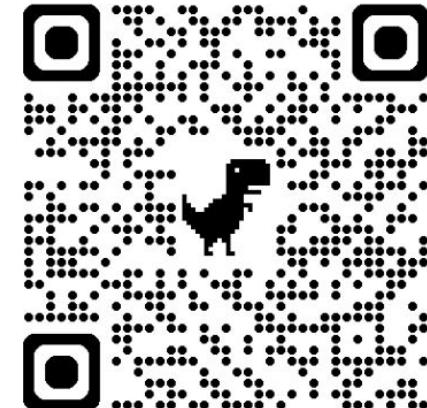


T.T. SOFTWARE SOLUTION

We are Software Solution Expert made by MVP Technical Team

เราคือผู้นำด้าน Microsoft Technology ในประเทศไทย
พัฒนาโซลูชันธุรกิจด้วย ASP.NET, Azure และ C# โดยทีม MVP และผู้เชี่ยวชาญ

สนใจติดต่อ 📞 086-899-6243 (สำนักงานกรุงเทพ) ✉️ nakorn@tt-ss.net T. T. Software Solution @ttss



ศึกษาความรู้เพิ่มเติม Generative AI (Learning Path)



Develop Generative AI solutions with Azure OpenAI Service

2 hr 13 min • Learning Path • 3 Modules

2600 XP

At a glance

Level
Intermediate

Product
Azure, Azure AI services, Azure SDKs

Role
AI Engineer, Developer

Subject
Artificial intelligence

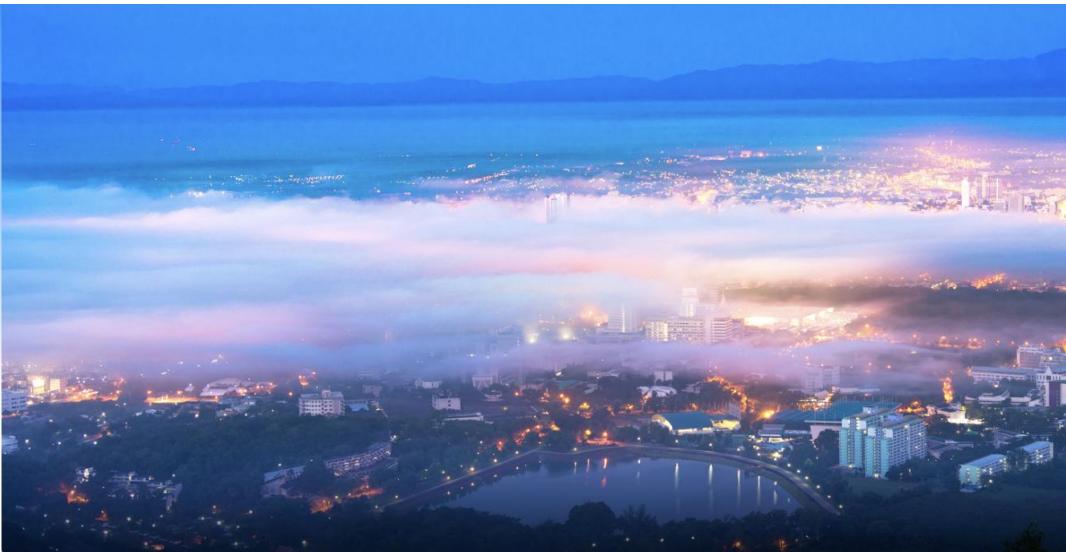
Azure OpenAI Service provides access to OpenAI's powerful large language models such as GPT and Embeddings models. These models enable various natural language processing (NLP) solutions to understand, converse, and generate content. Users can access the service through REST APIs, SDKs, and Azure AI Foundry.

Previous module



<https://learn.microsoft.com/en-us/training/path/develop-ai-solutions-azure-openai/>

ชุมชนโปรแกรมเมอร์หาดใหญ่ - สงขลา



ชุมชนโปรแกรมเมอร์หาดใหญ่ - สงขลา

Public group · 47 members



[Discussion](#)

[Events](#)

[Media](#)

[Files](#)

[People](#)