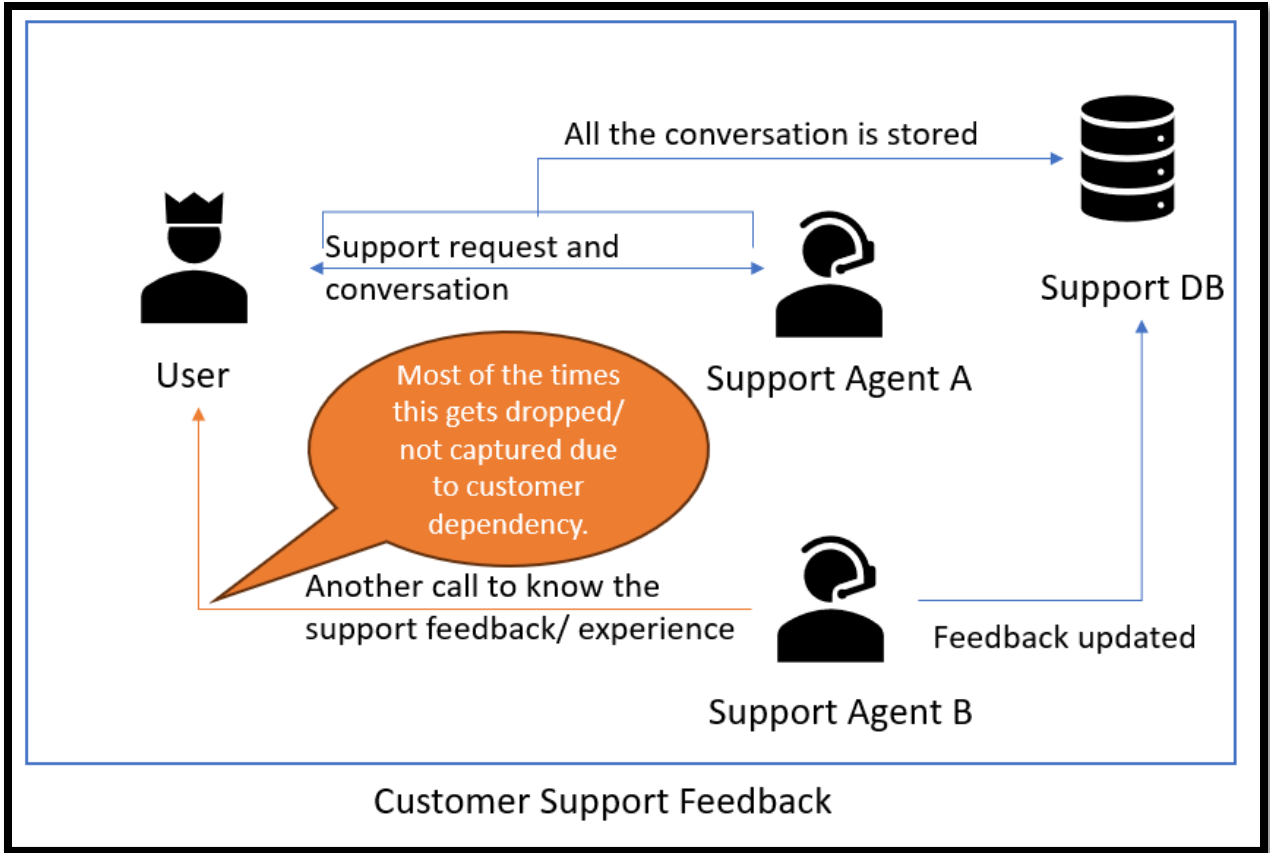


# Using OpenAI to Generate Customer Feedback

Within this blog post, I introduce a solution designed to facilitate the generation of customer feedback through the seamless integration of diverse tools available on the Microsoft Azure platform. The core objective of this approach is to optimize the feedback collection process by eliminating the requirement for supplementary steps, such as initiating phone calls to gather feedback following each support interaction. By leveraging the capabilities of Microsoft Azure, the solution seeks to create a more efficient and automated mechanism for obtaining valuable insights from customers. This not only enhances the overall workflow by reducing manual interventions but also contributes to a more agile and responsive feedback system. The integration of Azure tools not only aligns with modern technological advancements but also underscores the commitment to providing a customer-centric approach, ensuring that feedback is seamlessly woven into the support process without necessitating additional, time-consuming steps.

## Problem Statement :

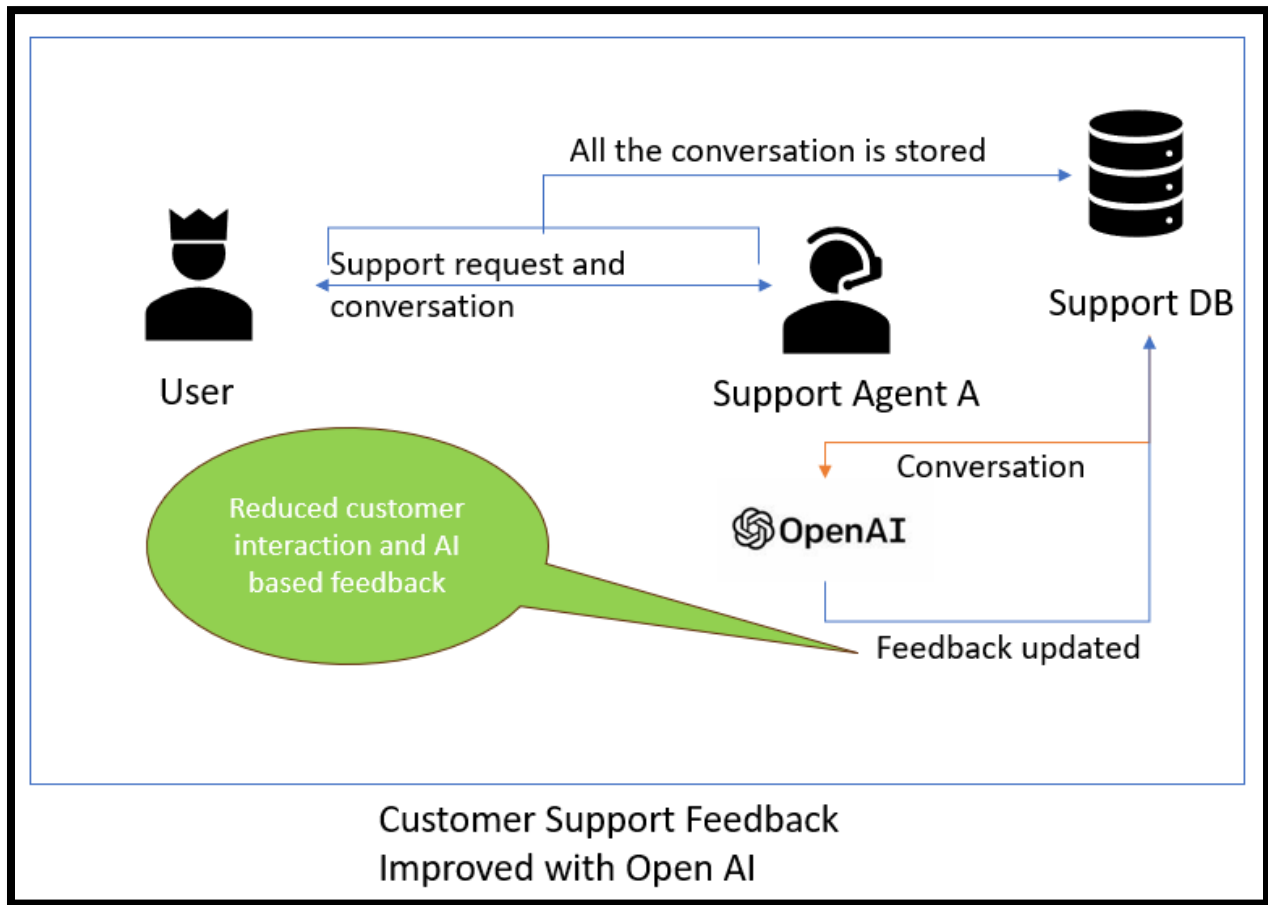
As per established support protocols, a standard operating procedure involves the systematic engagement of customers following a support conversation. This entails initiating additional communication, either through email or a follow-up call, to incorporate insightful feedback. The objective is to comprehensively assess customer satisfaction levels and pinpoint areas for improvement. This meticulous approach not only reinforces a commitment to customer-centricity but also serves as a valuable channel for addressing any concerns promptly. By integrating this feedback loop into the support process, we aim to enhance overall service quality and ensure a positive and proactive customer experience. And here is the pictorial representation of the problem statement:



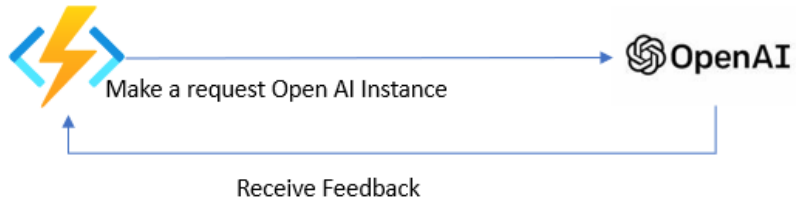
 Solution :

This proposed solution is designed to streamline our current operational workflow by eliminating a specific step in the process. Specifically, it aims to automate the generation of feedback for each support ticket. By implementing this solution, we intend to enhance efficiency and responsiveness in our support system. The automation process will prompt the system to proactively gather feedback without the need for manual intervention, thereby reducing the workload associated with this task. This enhancement not only simplifies the overall support procedure but also expedites the feedback collection process. Automated feedback generation is expected to provide valuable insights into customer satisfaction and areas that may require attention or improvement. Ultimately, this solution seeks to optimize our support operations, ensuring a more seamless and customer-centric approach while simultaneously empowering our team to focus on other critical aspects of customer service and issue resolution. And here is a

pictorial representation of the solution :



✉ Architecture Diagram:




## Using OpenAI to Generate Customer Feedback

Here are the steps to build it for your environment :

1. **Create an Azure AI Instance**

Step 1: Fill the basic details:

## Create Azure OpenAI ...

 Changes on this step may reset later selections you have made. Review all options prior to deployment.

**Basics** (1) 2 Network 3 Tags 4 Review + submit

Enable new business solutions with OpenAI's language generation capabilities powered by GPT-3 models. These models have been pretrained with trillions of words and can easily adapt to your scenario with a few short examples provided at inference. Apply them to numerous scenarios, from summarization to content and code generation.

[Learn more](#)

### Project Details

Subscription \* ⓘ

Resource group \* ⓘ  [Create new](#)

### Instance Details

Region ⓘ

Name \* ⓘ  ✓

Pricing tier \* ⓘ

[View full pricing details](#)

### Content review policy

To detect and mitigate harmful use of the Azure OpenAI Service, Microsoft logs the content you send to the Completions and image generations APIs as well as the content it sends back. If content is flagged by the service's filters, it may be reviewed by a Microsoft full-time employee.

[Learn more about how Microsoft processes, uses, and stores your data](#)

[Apply for modified content filters and abuse monitoring](#)

[Review the Azure OpenAI code of conduct](#)

Step 2: Fill the networking.

## Create Azure OpenAI ...

Basics
  **2 Network**
 3 Tags
  4 Review + submit

**i** Configure network security for your Azure AI services resource. 🗑️

Type \*

- All networks, including the internet, can access this resource.
- Selected networks, configure network security for your Azure AI services resource.
- Disabled, no networks can access this resource. You could configure private endpoint connections that will be the exclusive way to access this resource.

After creating, it will look like,

<input type="checkbox"/> Name ↑↓	Kind ↑↓	Location ↑↓	Custom Domain Name ↑↓	Pricing tier ↑↓	Status ↑↓	Created date ↑↓
<input type="checkbox"/>  az-openai-copilot-01	OpenAI	East US	 az-openai-copilot-01	S0	Succeeded	2023-12-08T13:22:26.715Z

## 2. Create a PowerShell Function

Step1 : Fill in the basic details.

# Create Function App ...

[Basics](#) [Storage](#) [Networking](#) [Monitoring](#) [Deployment](#) [Tags](#) [Review + create](#)

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

## Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Resource Group \* ⓘ   
[Create new](#)

## Instance Details

Function App name \* .azurewebsites.net

Do you want to deploy code or container image? \*  Code  Container Image

Runtime stack \*

Version \*

Region \*

## Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

Operating System \*  Linux  Windows

## Hosting

The plan you choose dictates how your app scales, what features are enabled, and how it is priced. [Learn more](#)

[Review + create](#) [< Previous](#) [Next : Storage >](#)

Step2: Add a http trigger:

## Create function ✕

**Select development environment**  
Instructions will vary based on your development environment. [Learn more](#)

Development environ... 🌐 Develop in portal

**Select a template**  
Use a template to create a function. Triggers describe the type of events that invoke your functions. [Learn more](#)

🔍 Filter

Template	Description
HTTP trigger	A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string
Timer trigger	A function that will be run on a specified schedule
Azure Queue Storage trigger	A function that will be run whenever a message is added to a specified Azure Storage queue
Azure Service Bus Queue trigger	A function that will be run whenever a message is added to a specified Service Bus queue
Azure Service Bus Topic trigger	A function that will be run whenever a message is added to the specified Service Bus topic
Azure Blob Storage trigger	A function that will be run whenever a blob is added to a specified container
Azure Event Hub trigger	A function that will be run whenever an event hub receives a new event

**Template details**  
We need more information to create the HTTP trigger function. [Learn more](#)

New Function\* collect

Authorization level\* ⓘ Function

Create
Cancel

Step 3: Add the bellow code:

```
using namespace System.Net
```

```
# Input bindings are passed in via param block.
param($Request, $TriggerMetadata)
```



```

$maxTokens=800
$textToSummarize=$Request.Body.FeedBackText
$apiKey="<api key>"

$uri = "<open ai api URL>"
$postIssueBody =@([pscustomobject]@{role = "system";content = "Could you please read the below conversation and provide a customer satisfaction result with any of the options as good, bad, worst " + $textToSummarize})
$headers = @{
    'api-key' = $apiKey
    'Content-Type' = 'application/json'
}
$body = @{
    messages = $postIssueBody
    max_tokens = 800
    temperature = 0.7
    frequency_penalty = 0
    presence_penalty = 0
    top_p = 0.95
    stop = $null
} | ConvertTo-Json

$parameters = @{
    Method      = 'POST'
    URI         = $uri
    Headers     = $headers
    Body        = $body
    ErrorAction = 'Stop'
}

$response = Invoke-RestMethod @parameters
$body =$response.choices[0].message.content
Write-Host $body

# Associate values to output bindings by calling 'Push-OutputBinding'.
Push-OutputBinding -Name Response -Value ([HttpResponseContext]@{
    StatusCode = [HttpStatusCode]::OK
    Body = $body
})

```

### 3. Collect the open ai api URL and key from the Open AI instance

Azure OpenAI

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Resource Management

- Keys and Endpoint
- Model deployments
- Encryption
- Pricing tier
- Networking
- Identity
- Cost analysis
- Properties
- Locks

Regenerate Key1 Regenerate Key2

**i** These keys are used to access your Azure AI service API. Do not share your keys. Store them securely— for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1  
..... ✓

KEY 2  
.....

Location/Region ⓘ  
eastus

Endpoint  
https://az-openai-copilot-01.openai.azure.com/ ✓

#### 4. Pass the conversation:

×

InputOutput

Provide parameters to test the HTTP request. Results can be found in the Output tab.

**HTTP method** ⓘ

POST▼

**Key**

master (Host key)▼

**Query**

---

[+ Add parameter](#)

---

**Headers**

---

[+ Add header](#)

---

**Body**

```
1  {"FeedBackText":"hi! i need to return an item, can you help me wit
```

Run

Close

5. View the Result :

Input **Output**

**HTTP response code**

200 OK

**HTTP response content**

Based on the provided conversation, it seems the agent is responsive and tries to assist the customer within the constraints of the return policy. The agent goes the extra mile by offering to escalate the issue to the manager and notifying them promptly. The customer expresses gratitude, and the agent maintains a polite and apologetic tone, even though they couldn't provide a resolution due to the policy.

Run

Close

## Custom configurations:

Tailoring this solution to your specific needs is easily achievable with a few customizable aspects.

- **Firstly**, you have the flexibility to select the language of your choice when crafting the Azure Function, allowing seamless integration into your existing language preferences.
- **Additionally**, you can fine-tune and optimize prompts to suit the unique requirements of your solution. This customization empowers you to align the solution with your distinct linguistic and functional preferences, ensuring a tailored and effective implementation that caters precisely to the nuances of your

```
$body = @{  
    prompt = "Could you please read the below conversation and provide a customer satisfaction result with any of the options as good, bad, worst: ""$textToSummarize"""
```

## Challenges faced :

Several significant challenges were encountered in the development of this solution.

- ✓ **Firstly, selecting the appropriate model in OpenAI posed a considerable hurdle.** After thorough research, it was determined that ChatGPT 3.5 was the most suitable choice, aligning seamlessly with the project's ease-of-use and efficiency requirements.
- ✓ **Another notable challenge involved crafting precise prompts and accurately capturing customer satisfaction levels (good, bad, worst).** This task demanded expertise and time, as formulating the right prompts is crucial for obtaining meaningful and accurate feedback. Achieving this required a nuanced understanding of language and customer sentiment to ensure the effectiveness of the feedback-generating process.
- ✓ **Additionally, the model's token limit posed a constraint.** Effectively breaking down the input within the token limit was imperative for optimal model performance. This challenge required careful consideration of input structure and content to maximize the efficiency of the solution.

In summary, overcoming these challenges involved meticulous research, language expertise, and strategic planning to ensure the seamless integration and effective functioning of the solution in generating valuable customer feedback.

## Business Benefits :

The implementation of this solution promises a myriad of business benefits, addressing critical aspects of operational efficiency and customer satisfaction.

- ✓ **Firstly, there is a substantial reduction in the total volume of customer calls.** By automating the feedback generation process, the need for extensive follow-up calls is minimized, allowing for a more streamlined and resource-efficient communication strategy.
- ✓ **Furthermore, the solution significantly diminishes customer interactions without compromising the quality of engagement.** This reduction in interactions contributes to an improved overall customer experience, as it streamlines communication, minimizing potential disruptions in the customer journey.
- ✓ **Moreover, a noteworthy benefit is the alleviation of workload and dependency on support executives.** With the automation of feedback collection, support teams can focus on more intricate and personalized customer queries, enhancing the overall efficiency of support operations.

- ✓ **Additionally, by choosing ChatGPT 3.5 as the underlying model, the solution ensures an optimal balance between ease of use and efficiency.** This model selection contributes to the solution's effectiveness in generating relevant and contextually appropriate feedback.

In conclusion, the implementation of this solution not only leads to resource optimization and enhanced operational efficiency but also nurtures a more positive and efficient customer experience, ultimately contributing to the overall success and effectiveness of the business.

**Reference :** [Azure OpenAI Service - Documentation, quickstarts, API reference - Azure AI services | Microsoft Learn](#)

**Published by :** [Using OpenAI to Generate Customer Feedback – Azure4you By Lalit Rawat](#)