

OpenAI (creators of ChatGPT) integration into Puzzel Case Management

The OpenAI integration into Puzzel Case Management facilitates increased automation of manual tasks that are both repetitive as well as uninteresting, resulting in a smoother agent experience and reduced waiting time for customers.

It is possible to create schemas for the following example scenarios, and trigger the request to ChatGPT with specific event rules relevant to the use case.

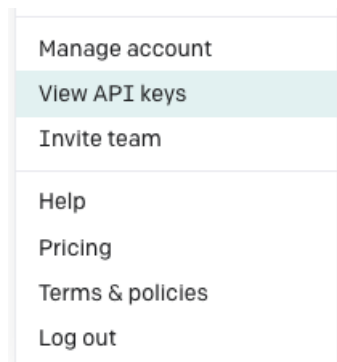
- Reduce manual agent tasks through suggested responses that can be automatically added as notes to tickets
- Automate the triage process with prioritisation tags that follow Puzzel Case Management's categorisation
- Contextualise the support request and facilitate the agents' understanding of requests, ensuring empathetic delivery through automatic generation of ticket notes that identify the customer sentiment in inbound messaging
- Less time lost for agents and reduced response times for clients, thanks to automated translations of messages into English
- Increase support efficiency during inbound peak times by automatically adding attributes to triage tickets when common business issues arise. This would then allow an agent to bulk respond with a template to these tickets.
- Clarity in categorisation with automatic ticket grouping using Puzzel Case Management's keywords

This guide outlines the steps needed to integrate OpenAI into Puzzel Case Management.

Steps

1. Sign-up for an account at openai.com Once you have created the account

- Click on your profile icon at the top-right hand side
- Select View API Keys
- Create a new secret key (note down the API key as you will need it in a future step).

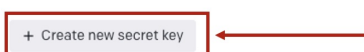


API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED	
sk-...9PHo	27 Feb 2023	Never	



2. Sign-up for an account with Zapier (zapier.com)

You can use Zapier as a web-service tool to accept the receiving webhook from Puzzel Case Management, communicate with the OpenAI API and then return the response back into Puzzel Case Management's API to update the ticket.

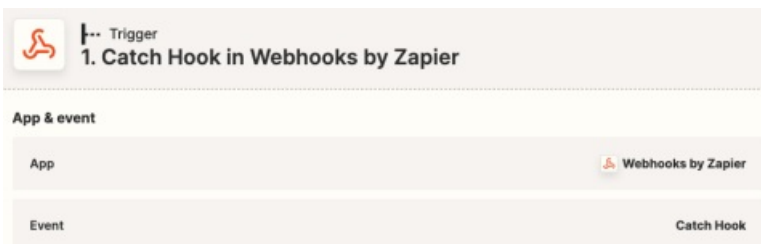
Alternatively, you can ask your own developer to build a custom web-service application for this operation.

Below you can find an example flow that can be easily built using Zapier.



When configuring Step One of your Zap, you should use the following settings (please see screenshot below).

It is important to note down the webhook URL which is generated, as this will be needed in a future step to complete the integration.



3. Then you will need to **access Puzzel Case Management** and:

- Navigate to Settings
- Click Integrations
- Select Outbound Integrations
- Click Webhooks
- Create a new webhook
- Paste the webhook URL from the previous step

URL*

Name*

Method*

Authenticates Via*

Custom headers

Name	Value *	
<input type="text" value="New Attribute Name"/> ?	<input type="text" value="New Attribute Value"/> ?	<input type="button" value="+"/>

4. In Puzzel Case Management navigate to

- Settings
- Then Integrations
- Click on Outbound Integrations
- Select Schema
- Create a new Schema
- Select content type 'JSON'

Below you will find suggested use cases.

Schema which will prompt OpenAI to return a suggested agent response to the most recently received message content in the ticket.

Schema Details

Name

Suggest an agent message response

Content Type

JSON

Content

```
{  
  "ticket_ref": "{{ticket:ticket_ref}}",  
  "message": "Give me a suggested response for the following message I have received. - {{ticket:recent_content}}."  
}
```

Schema will prompt OpenAI to prioritise the ticket based on the most recent inbound message.

Schema Details

Name

Prioritise the message

Content Type

JSON

Content

```
{  
  "ticket_ref": "{{ticket:ticket_ref}}",  
  "message": "How should I prioritise the following message for my agent to handle? Please rate it in one of the following categories  
junk, lowest, low, normal, high, highest - {{ticket:recent_content}} - Please respond with only one word"  
}
```

5. The next step is to click on:

- Build an Event Rule
- Then navigate to Settings
- Click on Productivity
- Then select Event Rules
- Create a new rule. This will trigger the Schema to be sent to the webhook that you have already created in previous steps. You could use a ticket event (i.e.: "message received" or "tag added"). For further information about how to create and manage Event Rules you will need to refer to the article on [Event Rules](#).

6. Test the first step of your Zapier flow. If the event rule successfully triggered, you should see your payload appear in Zapier.

7. In **Step 2 of the Zap**, you should select the options shown in the screenshot below. You will then be asked to enter the API key you obtained from step 1 of this article.

OpenAI (Makers of ChatGPT) Beta Change

* Event (required)
Send Prompt

1. This is performed when the Zap runs.

In the actions section, you should use the following (please see screenshot below). All other settings should be left with the default values in place.

* Model (required)
text-davinci-003

We recommend "text-davinci-003". For a faster and lower cost option try "text-curie-001". [Learn More.](#)

* Prompt (required)
1. Message: How should I re...lp@puzzel.com

You can then test the action. (You should see a sample response returned in Zapier).

8. **Step 3 of the Zap** requires the app called 'Code by Zapier'. This is needed to cleanse some of the control characters from the response.

3. Run Javascript in Code by Zapier

Choose app & event ✓

Code by Zapier Change

* Event (required)
Run Javascript

When setting up the action, you should use the following:

^ Set up action ✔

Input Data

filteredoutput

2. Choices Text: The capital o...d is Edinburgh.



And enter the following code:

```
// Define the output variable
var output = {}; // Get the value of the text input
var text = inputData.filteredoutput; // Remove any line breaks from the text
text = text.replace(/(\r\n|\n|\r)/gm, ""); // Set the output variable to the processed text
output.filteredoutput = text; // Log the output to the console
console.log(output); // Return the output variable
output;
```

Then test that the action works correctly.

9. Then you will need to access **Puzzel Case Management** and do the following:

- Navigate to Settings
- Select Ticket Channels
- Click on API
- Create a new 'Global Authentication'
- Note down the API key as you will need it for the following step

API Details

Name
Puzzel


Authentication Level
Global

Token Type
Basic Token


Authentication
Refresh API Access

API Key [Show Token](#)

10. The 4th step of the Zap requires you to use the app 'Webhooks by Zapier' and select the event 'Custom Request'.

 **Action**
4. Custom Request in Webhooks by Zapier ⋮

^ Choose app & event ✓

 **Webhooks by Zapier** **Premium** Change

* Event (required)

Custom Request ⌵

This is performed when the Zap runs.

Continue

The action should be setup as follows:


Set up action ✓

* Method (required)

PUT ⌵

The HTTP method we'll use to perform the request.

* URL (required)

<https://tenantname.logicalware.com/api/v1/tickets/>  1. Ticket Ref: 595

Any URL with a querystring will be re-encoded properly.

Data Pass-Through?

False ⌵

If yes, request [data](#) below will be overridden with the raw data from the previous step (you are still responsible for... [more](#))

Data

```
{
  "notes": [
    {
      "body": "</> 3. Filteredoutput: The capital of...d is Edinburgh. "
    }
  ]
}
```

Place raw data for the request here. Could be JSON, XML, etc. Nothing is parsed for you here (including extra line... [more](#))

Unflatten

By default we convert fields with double underscores into nested dictionaries. Select no here to turn this behavior off.

Basic Auth

A pipe (|) separated username, password combo for standard HTTP authentication.

Headers

X-Api-Token	secrettokenfrompuzzelcasemanagement	X
Content-Type	application/json	X
		+

Key, value pairs to be added as headers in all requests.

Testing this action should update the referenced ticket in Puzzel Case Management by inserting your filtered output as a note into the ticket.

Warning

'Event Rules' and 'API Channel' should be enabled on your Puzzel Case Management account in order to complete all of the above steps. If these features are not currently enabled on your Case Management instance, please raise a case with our support team.