RK-A2 - Feature #3772

Colab and GCF changes to use OpenAI chat endpoint

11/26/2024 07:24 AM - Ram Kordale

Status:	Closed	**Start date:**	11/26/2024
Priority:	Urgent	**Due date:**	
Assignee:	Ram Kordale	**% Done:**	100%
Category:		**Estimated time:**	1.00 hour
Target version:		**Spent time:**	0.00 hour

Description

3771 colab uses our GCF that uses the OpenAI Assistants end point (https://platform.openai.com/docs/api-reference/assistants/createAssistant). Clone 3771 colab to ensure the new colab does all that 3771 does except that it uses a new GCF (to be built as part of this ticket) that uses the OpenAI Chat endpoint (https://platform.openai.com/docs/api-reference/chat/create).

However, the support required is minimal and is described below.

POST call needs to support only 2 inputs:
- model
- messages. However, we will not have "system" role messages. We will only have "user" and "assistant" role messages. This call returns a 'chat completion' object and not a streamed sequence. We need to process only three fields from the response:
--id: just print this.
--choices array. Process only the first choice. Print the 'finish_reason' and retrieve the "content" of the message for further processing as this is the actual response to our prompt.
--usage: please print the fields.

The above doc page (https://platform.openai.com/docs/api-reference/chat/create) contains samples that can be used for dummy testing.

The chat endpoint does not have an equivalent of threads (and therefore thread_id). Instead, replace thread_id with a string variable thread.

So, wherever thread_id is "" in 3771 colab, make thread's value "new". After making the openai call, make thread's value as "old".

When thread's value is "new", the messages input should have only one part. See example message in the input below: {
"model": "<model>",
"messages": [{
"role": "user",
"content": [{
"type": "text",
"text": "<prompt-1>"
}
]
}
]
}

and the message in the response will contain: {
...
"message": {
"role": "assistant",
"content": "<response-1>"
}
...
}

When the value of threads is "old", messages should contain the chat so far. So, when we continue the above example configuration,
{
"model": "<model>",
"messages": [{
"role": "user",
"content": [{

"type": "text",
"text": "<prompt-1>"
}
]
}, {
"role": "assistant",
"content": [{
"type": "text",
"text": "<response-1>"
}
]
}, {
"role": "user",
"content": [{
"type": "text",
"text": "<prompt-2>"
}
]
}
]
}

and the message in the response will contain: {
...
"message": {
"role": "assistant",
"content": "<response-2>"
}
...
}

History

#1 - 11/29/2024 04:59 PM - Shubham Boke

- Status changed from New to In Progress

- Assignee set to Shubham Boke

- % Done changed from 0 to 100

#2 - 11/29/2024 04:59 PM - Shubham Boke

- Status changed from In Progress to Resolved

#3 - 11/29/2024 04:59 PM - Shubham Boke

- Status changed from Resolved to Review

- Assignee changed from Shubham Boke to Parag Patil

#4 - 12/06/2024 12:33 PM - Dewakar Chaubey

- Status changed from Review to Feedback

- Assignee changed from Parag Patil to Shubham Boke

#5 - 01/06/2025 05:16 AM - Ram Kordale

- Estimated time set to 1.00 h

#6 - 02/06/2025 12:17 PM - Shubham Boke

- Assignee changed from Shubham Boke to Parag Patil

#7 - 02/06/2025 12:44 PM - Dewakar Chaubey

- Assignee changed from Parag Patil to Ram Kordale

#8 - 03/27/2025 05:50 AM - Ram Kordale

- Status changed from Feedback to Closed