Infosys

Responsible AI Office

Infosys Responsible AI Toolkit

Hallucination

API usage Instructions

Contents

About Hallucination

Hallucination in large language models (LLMs) refers to the generation of information that appears plausible but is factually incorrect, fabricated, or misleading. This occurs because LLMs predict text based on patterns in the data they were trained on, rather than verifying facts against a database or external reality. Hallucinations are especially important to address because they can erode user trust, spread misinformation, and lead to harmful or costly outcomes in high-stakes applications such as healthcare, law, or education. Reducing hallucinations is a key challenge in making LLMs more reliable and safe for real-world use.

Once API swagger page is populated as per instructions given in the github repository Readme file, click on 'try it out' to use required endpoints. Details of endpoints associated with Halluciantion tenet are outlined below.

Dependencies

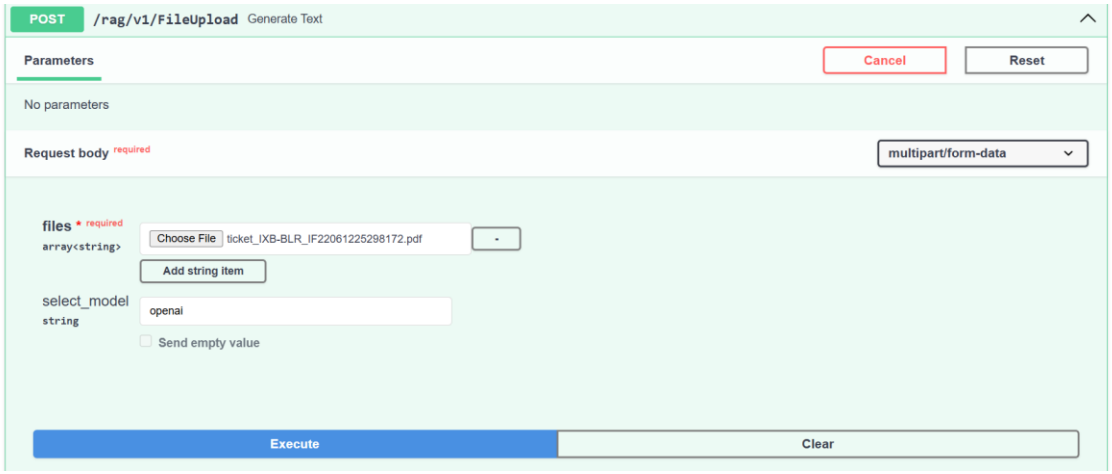
Please follow the setup instructions in the README files of the hallucination repository to know about the dependencies and installation procedures.

Features & API Endpoints:

1. FileUpload -->

It is used to upload document with pdf/csv/txt format and the llm model (openai/Gemini/aws) and returns the vectorestoreid of vectorestore created.

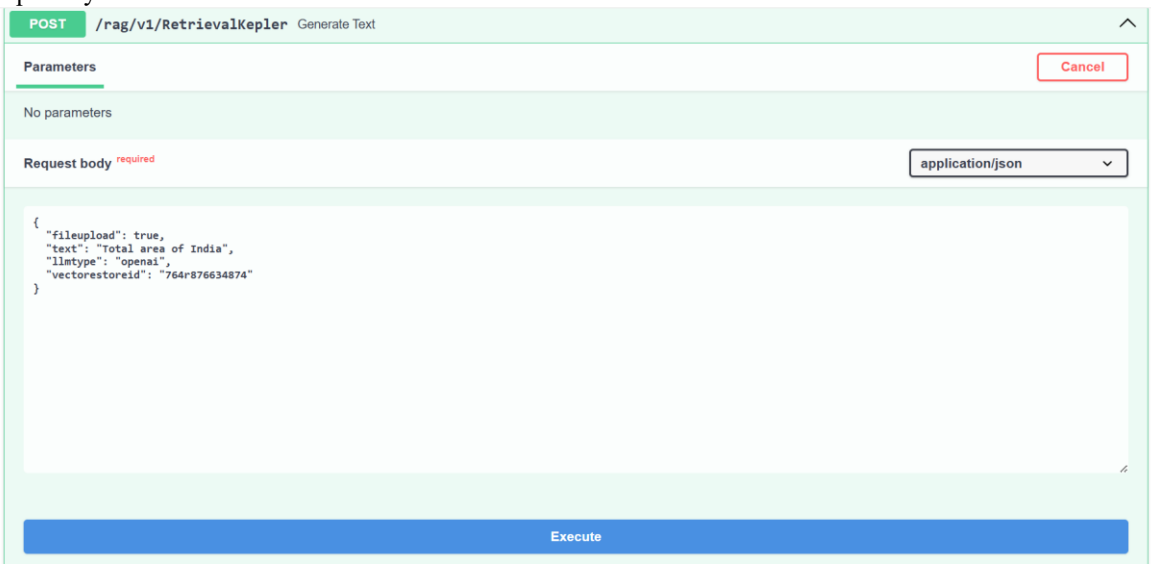
Input Payload:



2. RetrievalKepler -->

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), prompt, vectorestoreid and llm model (openai/Gemini/aws) as Input and will return a response along with the score.

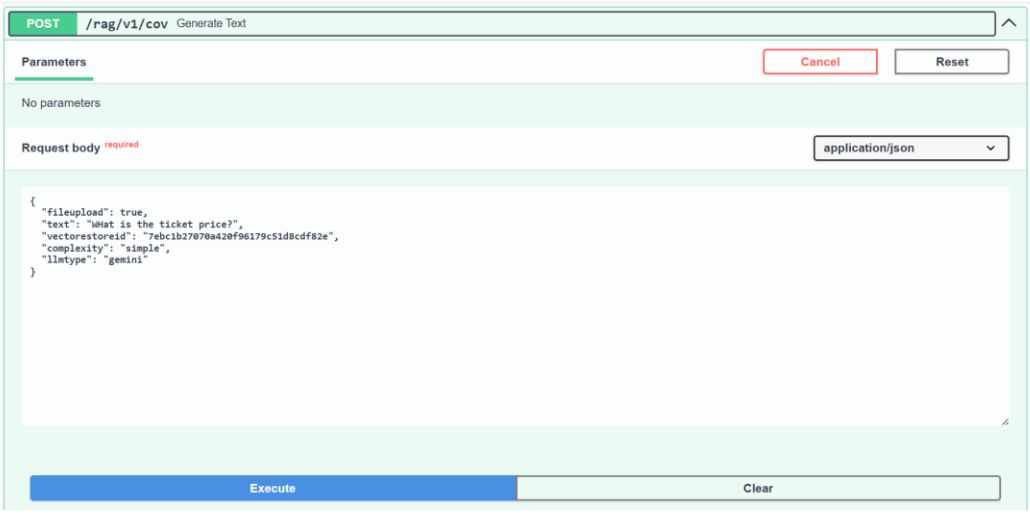
Input Payload:



3. cov – ->

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), text, vectorestoreid and llm model (openai/Gemini/aws) as Input and will return the response with 5 more variants of questions generated by LLM to verify the answer and a refined final response.

Input Payload :

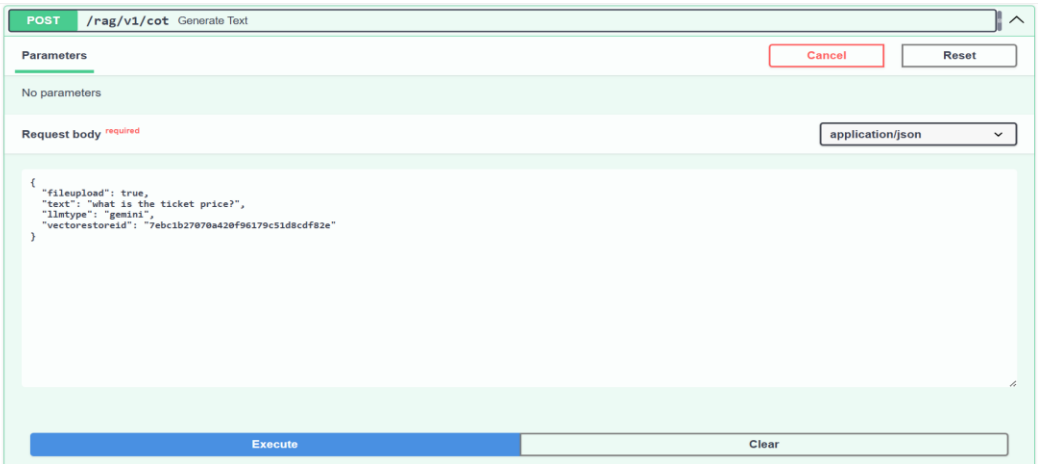


4. Chain of thought (CoT)-

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), text, vectorestoreid and llm model (openai/Gemini/aws) as Input and will return the response. Chain of Thought response provides explanation steps and reasoning behind and from

where (source).

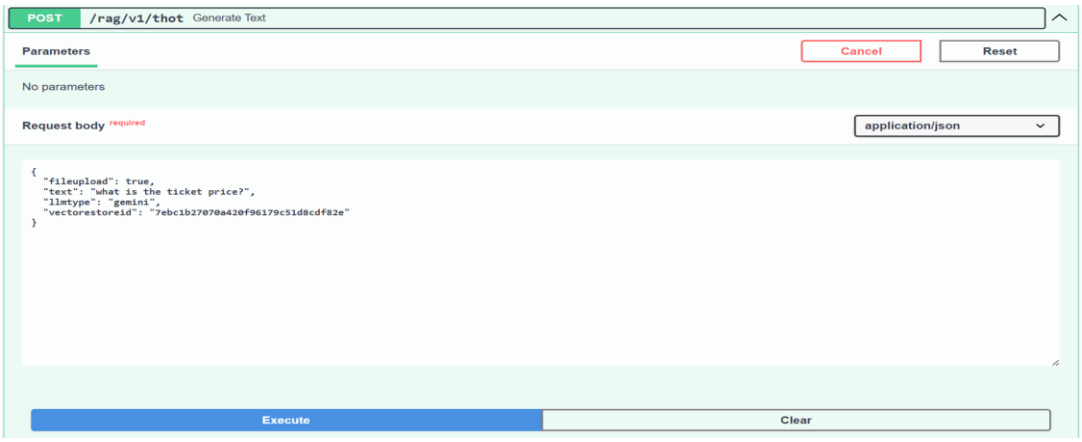
Input Payload:



5. Thread of Thought(ThoT)-

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), text, vectorestoreid and llm model (openai/Gemini/aws) as Input and will return Thread of Thought response. Thot is efficient in more descriptive and complex information spread over the file.

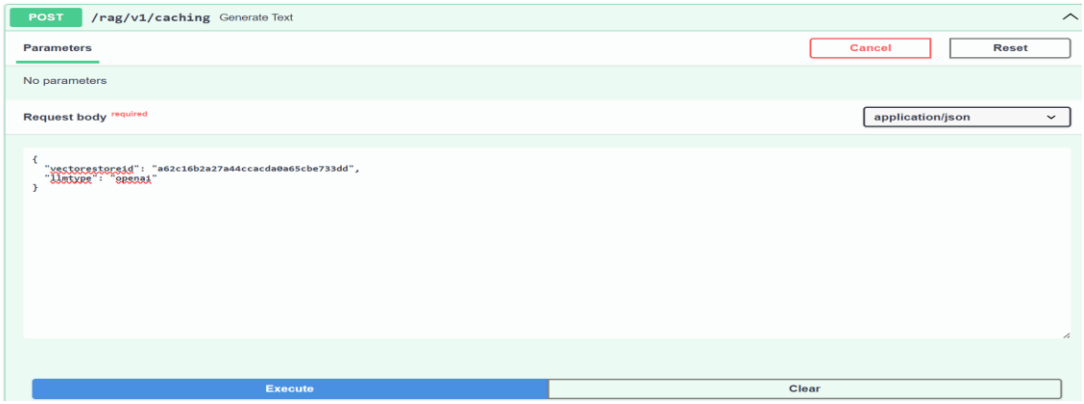
Input Payload :



6. Caching-

This will take vectorstoreid which is generated while uploading file and and llm model (openai/Gemini/aws) and will return an array of length 2. First element will be the Cache id, Second element will be the Cache id which is removed from the cache if cache is full else 0.

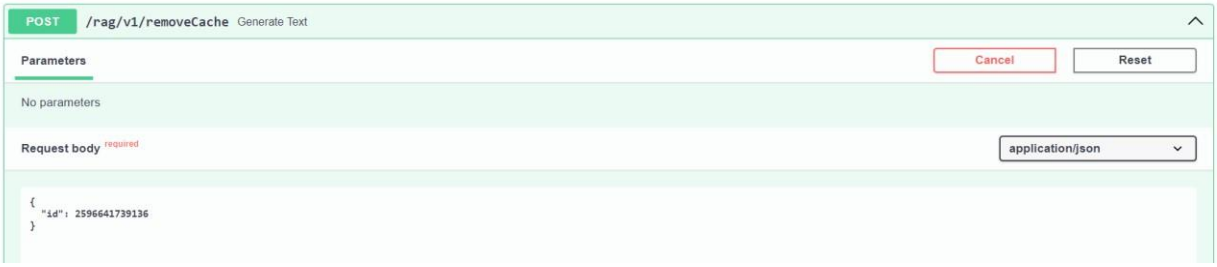
Input Payload:



7. RemoveCache-

It will take input as cache id and will remove the cached Vectorestore from the Cache.

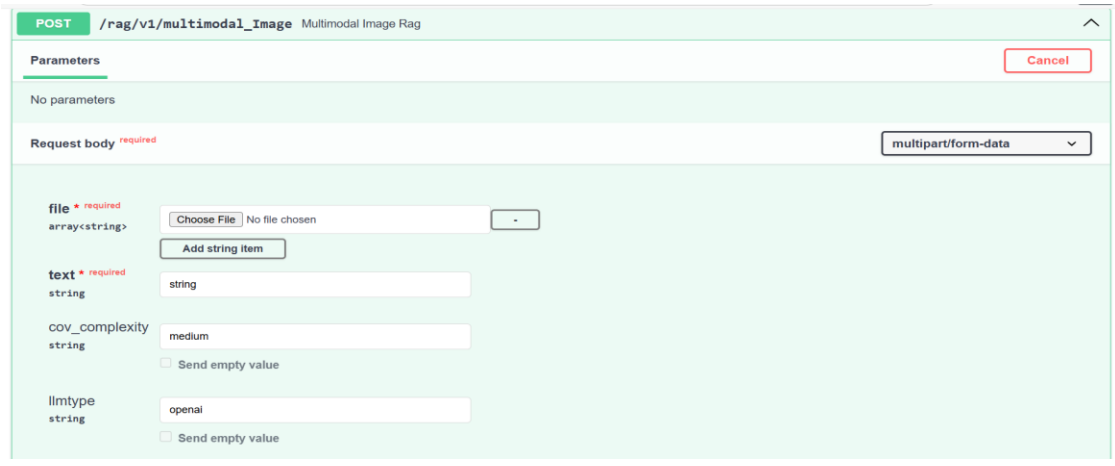
Input Payload:



8. Multimodal Image –

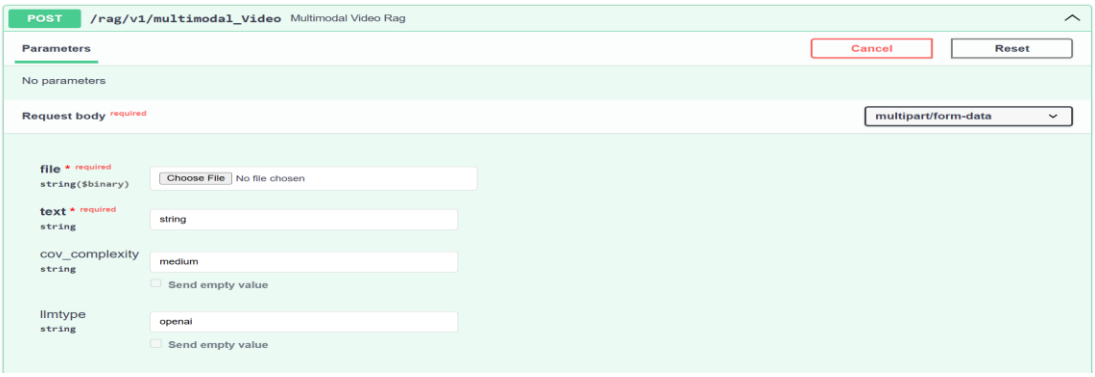
This will take images, prompt and chain of verification complexity and llm model (openai/Gemini/aws) as input and give corresponding response along with detailed explanations and hallucination scores.

Input Payload:



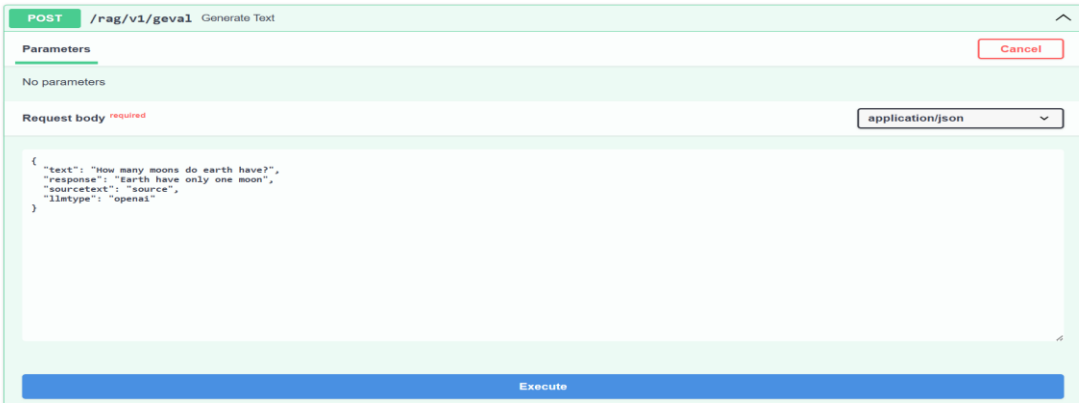
9. Multimodal Video –

This will take video, prompt and Cove complexity and llm model (openai/Gemini/aws) as input and give corresponding response along with detailed explanations and hallucination scores.
Input Payload:



10. Geval-

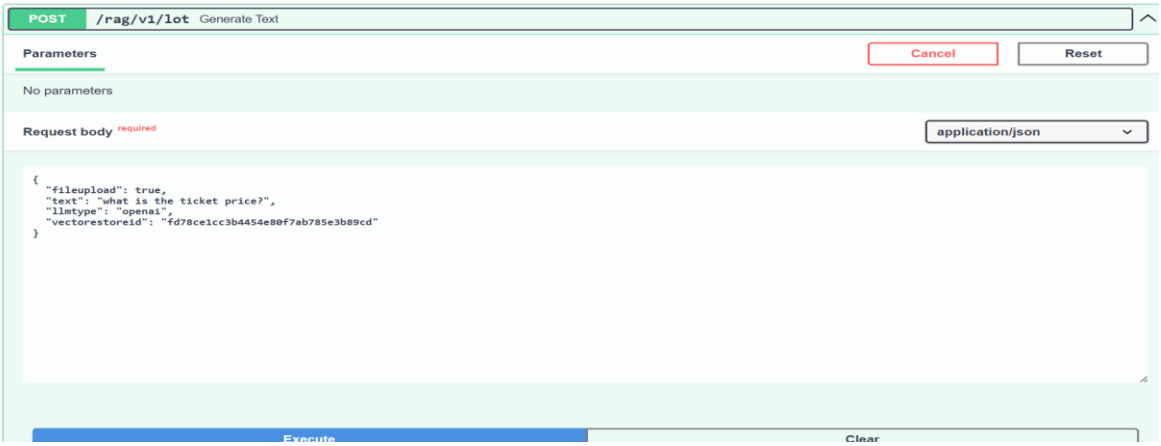
This will take as input the prompt, the corresponding response the source and llm model (openai/Gemini/aws) and generates scores along with explanation for Geval parameters (adherence, correctness, faithfulness and relevance) based on the prompt, response and source.
Input Payload:



11. Logic of Thought(LoT)-

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), text, vectorestoreid and llm model (openai/Gemini/aws) as Input and will return Logic of Thought response. Lot is efficient in more logical and complex information.

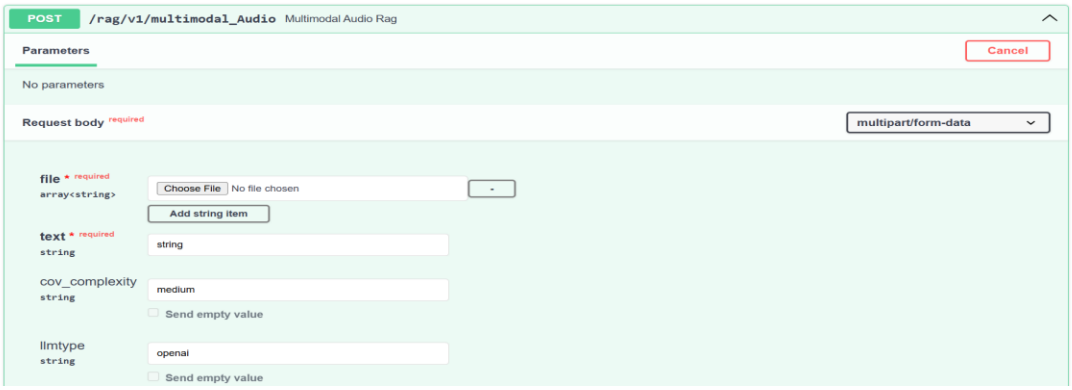
Input Payload :



12. Multimodal Audio –

This will take audio file, prompt, chain of verification complexity and llm model (openai/Gemini/aws) as input and give corresponding response along with detailed explanations and hallucination scores.

Input Payload:



Endpoints Usage Flow

Endpoint	Description
/FileUpload	Upload files in pdf/csv/txt format and returns vectorstoreid.
/RetrievalKeplar	Takes prompt along with vectorstoreid for the uploaded file to generate response along with hallucination score among other details.
/cov	Generates chain of verification response.

/cot	Generates chain of thoughts response.
/thot	Generates thread of thoughts response.
/lot	Generates logic of thoughts response.
/cache	Creates cache for files.
/removecache	Removes a particular cache file
/geval	Generates scores along with explanation for Geval parameters based on the prompt, response and source.
/multimodal_Image	Performs RAG for image file along with detailed explanations and hallucination scores.
/multimodal_Video	Performs RAG for video file along with detailed explanations and hallucination scores.
/ multimodal_Audio	Performs RAG for audio file along with detailed explanations and hallucination scores.