

## RAG

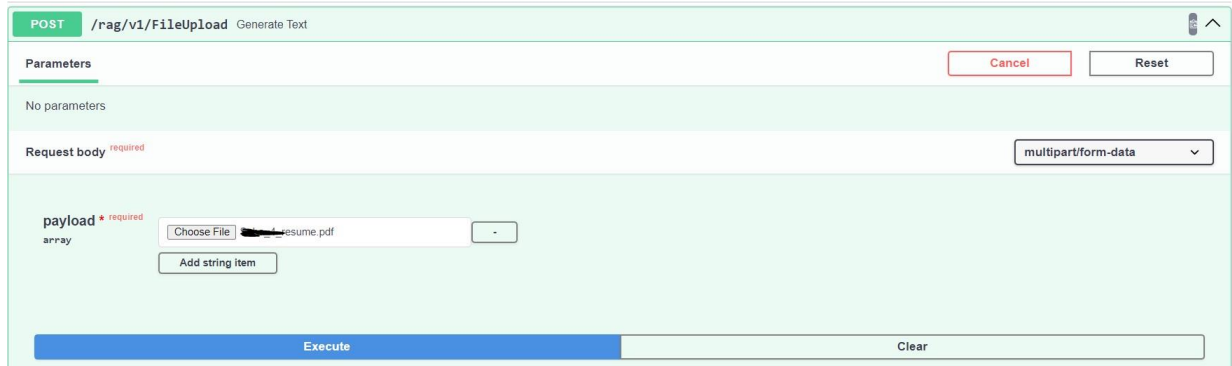
To try the endpoints, click on “try it out”

Endpoints:

### 1. FileUpload -->

It is used to upload document with pdf format and return vectorestoreid of vectorestore and blobname for the pdf.

Input Payload:



POST /rag/v1/FileUpload Generate Text

Parameters

No parameters

Request body <sup>required</sup>

multipart/form-data

payload <sup>required</sup>  
array

Choose File resume.pdf

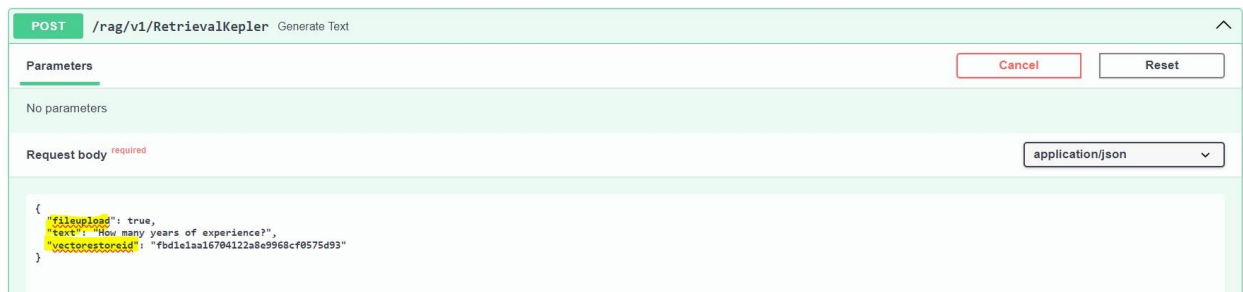
Add string item

Execute Clear

### 2. RetrievalKepler -->

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), text, vectorestoreid as Input and will return a rag response along with the score.

Input Payload:



POST /rag/v1/RetrievalKepler Generate Text

Parameters

No parameters

Request body <sup>required</sup>

application/json

```
{
  "fileupload": true,
  "text": "How many years of experience?",
  "vectorestoreid": "fbd1e1aa16704122a8e9968cf0575d93"
}
```

### 3. CoVe -->

This will take keys FileUpload(True if using File Upload else False when using Vectorestore Caching), text, and vectorestoreid 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 :

POST /rag/v1/cov Generate Text

Parameters

No parameters

Request body required application/json

```
{
  "fileupload": true,
  "text": "How many years of experience?",
  "vectorstoreid": "fbd1e1aa16704122a8e9968cf0575d93",
  "complexity": "simple"
}
```

#### 4. Chain of thought (CoT)-

This will take keys FileUpload(True if using File Upload else False when using Vectorstore Caching), text, vectorstoreid as Input and will return the response.

Chain of Thought response provides you explanation steps and reasoning behind and from where (source referred which document) Input Payload:

POST /rag/v1/cot Generate Text

Parameters

No parameters

Request body required application/json

```
{
  "fileupload": true,
  "text": "How many years of experience?",
  "vectorstoreid": "fbd1e1aa16704122a8e9968cf0575d93"
}
```

#### 5. Thread of Thought(ThoT)-

This will take keys FileUpload(True if using File Upload else False when using Vectorstore Caching), text, vectorstoreid as Input and will return Chain of verification response. That is efficient in more descriptive and complex information spread over the file. Input Payload :

POST /rag/v1/thot Generate Text

Parameters

No parameters

Request body required application/json

```
{
  "fileupload": true,
  "text": "How many years of experience?",
  "vectorstoreid": "fbd1e1aa16704122a8e9968cf0575d93"
}
```

#### 6. Caching-

This will take blobname which is generated while uploading file and return a 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:

POST /rag/v1/caching Generate Text

Parameters

No parameters

Request body **required** application/json

```
{  "blobname": {    "saba_d_resume_091c46bc-3d04-4dba-8e34-1f7505914c3d.pdf"  }}
```

## 7. checkCache -

It will show the current cache ID.

## 8. RemoveCache-

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

## Input Payload:

POST /rag/v1/removeCache Generate Text

Parameters

No parameters

Request body **required** application/json

```
{  "id": 2596641739136}
```

## 9. Show\_Score –

This will take input as Prompt, response which RetrievalKepler gave, SourceArr which is nothing but page content from RetrievalKepler as Input and Will return Hallucination Score.

## Input Payload:

POST /rag/v1/Show\_Score Generate Text

Parameters

No parameters

Request body **required** application/json

```
{  "prompt": "year of experience",  "response": "4 years of experience in relevant fields",  "sourcearr": [    {      "Summary": [        "4 years of experience in analysis, python, Machine Learning, Predictive Analysis, NLP and Deep Learning, Devops, AWS -EKS/S3, CI/CD.",        "application security vulnerabilities like application service container vulnerabilities, SAST, SonarQube.",        "Created web API's using python in Django framework.",        "Dockerized application micro services.",        "Developed document extraction POCs using our AI Application.",        "Created Swagger for application in Django for consumption by external clients to know API details.",        "Experience in working with multiple vendors and multiple teams.",        "Key Domain and Technical Knowledge",        "Domain : Data Science, AI, Cloud, DEVOPS, Automation",        "Technical : Python, Machine Learning, NLP, Deep Learning, Database - MySQL, MongoDB, Docker, AWS - S3, EKS, Texttrac, RPA -Assist Edge, GENAI (learning)",        "Total Work Experience:",        "Company: Infosys",        "Period : April 2022 to Present",        "Project Title: IECOP Homegrown AI Product",        "Description : IECOP (Infosys Enterprise Cognitive Platform), this platform provides API based services for multiple use case(s) including image, voice, video, OCR, digital automation, text analytics by combining application of ML, Cognitive, NLP, and AI principles and technique.",        "Role and Responsibilities:",        "Collaborated with a prestigious investment management company to implement advance document extraction capabilities using AI product.",        "Developed document extraction POCs using IECOP AI product using frameworks Django, PyTorch, TensorFlow, Keras.",        "Employed cutting -edge technologies including OCR (Optical Character Recognition) to atleast of 80% as per SONAR policies.",        "Creation Low level design documents which include class and sequence diagram.",        "Also, use-case based PDD & SDD",        "Participated in code and design reviews for code quality improvements ensuring a smooth transition between development, testing, deployment phases.",        "Munshi Saba Farheen",        "Contact Number: 9347754957",        "Email id: saba.munshi97@gmail.com",        "Current Location: Hyderabad",        "LinkedIn : linkedin.com/in/saba-farheen-munshi-a85a98149",        "Role Designation: Associate Consultant relationships.",        "Gathered Data from RDS and Redshift, applied Data Cleaning and Feature Engineering over data.",        "Implemented various time series forecasting techniques to predict alerts caused by refrigeration failure.",        "Develop action plans to mitigate risks in decision making while increasing profitability by Leveraging Data Science.",        "Company: Pantech Solutions Pvt L td.",        "Period : Jan 2020 to Aug 2021",        "Role and Responsibilities :",        "Handled 24x7 engineering requests in the Machine Learning domain.",        "Performed system analysis, documentation, testing, implementation.",        "Academic Qualification : Master's"      ]    }  ]}
```

Execute Clear

## 10. Multimodal Image –

This will take images, prompt and chain of verification complexity as input and give corresponding response.

Input Payload:

The screenshot shows the API interface for the endpoint `POST /rag/v1/multimodal_image`. The interface includes a "Parameters" section with "Cancel" and "Reset" buttons. Below this, the "Request body" is configured as "multipart/form-data". The input fields are:

- file** (required, array): A file selection button labeled "Choose File" with the filename "testimage3.jpg".
- text** (required, string): A text input field containing the prompt "what are the characters in the image?".
- cov\_complexity** (string): A dropdown menu set to "medium".

There is a checkbox for "Send empty value" which is currently unchecked. An "Execute" button is located at the bottom of the form.

## 11. Multimodal Video –

This will video and prompt as input and give corresponding response.

Input Payload:

The screenshot shows the API interface for the endpoint `POST /rag/v1/multimodal_video`. The interface includes a "Parameters" section with "Cancel" and "Reset" buttons. Below this, the "Request body" is configured as "multipart/form-data". The input fields are:

- file** (required, string(\$binary)): A file selection button labeled "Choose File" with the filename "Privacy\_Audio.mp4".
- text** (required, string): A text input field containing the prompt "What is the video about?".

An "Execute" button is located at the bottom of the form.

## 12. Geval-

This will take as input the prompt, the corresponding response and the source and show the geval metrics score along with explanation.

Input Payload:

POST

/rag/v1/geval

Generate Text

⌵

Parameters

Cancel

No parameters

Request body required

application/json

⌵

```
{
  "text": "How many moons do earth have?",
  "response": "Earth have only one moon",
  "sourcetext": "source"
}
```

Execute