

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:

The screenshot shows a REST client interface for the `POST /rag/v1/FileUpload` endpoint. The interface includes a "Parameters" section with "No parameters" listed. The "Request body" section is set to "multipart/form-data" and shows a "payload" array with one item: a file named "Resume.pdf". The "Execute" button is highlighted in blue.

2. Retrieval -->

This will take the id (vectorestoreid returned by FileUpload), and text(Prompt) as Input and will return the rag response by searching the document along with hallucination score and source of document.

Input Payload:

The screenshot shows a REST client interface for the `POST /rag/v1/Retrieval` endpoint. The interface includes a "Parameters" section with "No parameters" listed. The "Request body" section is set to "application/json" and shows a JSON payload:

```
{  "text": "How many years of experience?",  "id": "fbd1e13a16704122a8e996dcf8575d93"}
```

3. 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 Cancel Reset

No parameters

Request body required application/json

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

4. CoVe -->

This will take keys FileUpload(True if using File Upload else False when using Vectorstore Caching), text, and vectorstoreid 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 Cancel Reset

No parameters

Request body required application/json

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

5. 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 Cancel Reset

No parameters

Request body required application/json

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

6. Thread of Thought(ThoT)-

This will take keys FileUpload(True if using File Upload else False when using Vectorstore Caching), text, vectorestoreid 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?",  "vectorestoreid": "fbd1e1aa16704122a8e9968cf0575d93"}
```

7. 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-3d84-4dba-8e34-1f7505914c3d.pdf"  ]}
```

8. RemoveCache-

It will take input as cache id and will remove the cached Vectorstore 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

Cancel

Reset

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, Experience in identifying and fixing application security vulnerabilities like application service container vulnerabilities, SAST, SonarQube, Created web APK'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, Textract, RPA -Assist Edge, GENAI (Learning) Total Work Experience: Technical : Python, Machine Learning, NLP, Deep Learning, Database - MySQL, MongoDB, Docker, AWS - S3, EKS, Textract, RPA -Assist Edge, GENAI (Learning) Company: Infosys Period: April 2022 to Present
Project Title: IECP Homegrown AI Product Description: IECP (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 IECP 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: Download the engineering projects in the Machine Learning domain Performed system analysis, documentation, testing, implementation, Academic Qualification : Master's

Execute

Clear