

## RAG - Retrieval Augmented Generation

LLMs are trained only with data and LLMs are limited by the recency of data. How good the data is, that good is only LLM.

Also, lets say we have a chatbot and it has to respond only from the data we have with us, not something very generic

User ---> sends request ---> LLM

LLM is trained on some generic internet data but we want LLM to respond based on our data only

1. One option is to fine-tune LLM to make it respond in the way we want. We can train LLM more to respond to our data
2. Second option is send the entire dataset for every request but the problem is, we cant keep sending TBs of data for every request, it could consume bandwidth, computation, token cost. Most important is cost based on the tokens we are consuming from the data
3. Third option is, RAG. Reference documents could be PDF or docs, you dont want to send the entire private data to LLM (that's insecure)

PDFs are converted into multiple chunks, then Embeddings and stored in VectorStore. When user sends a query it goes to VectorStore. VectorStore returns matches. So the User query + matches go into LLM and LLM responds to User finally.

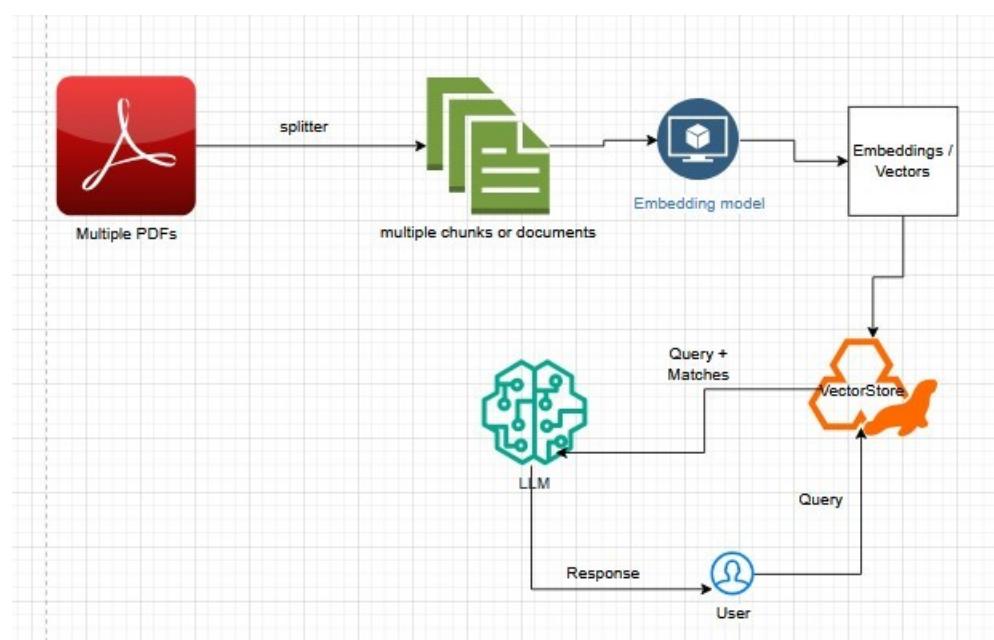
How do we include our documents into LLM?

PDF/docs ---> splitter ---> multiple chunks or documents ---> Embedding model ---->  
Embeddings/Vectors ---> VectorStore

User ---> Query ---> VectorStore (SimilaritySearch) ---> matches + query ---> LLM

LLM ---> responds ---> User

Check RAG\_Concept pic



// you are talking to LLM but you are getting response based on your Product list  
// for RAG

The screenshot shows a REST API testing interface. At the top, there is a header bar with a 'Save' button, a 'Share' button, and a 'Send' button. Below the header, the URL 'http://localhost:8080/api/ask/suggest products for vacation' is entered into a search bar. The method dropdown shows 'GET'. The main interface has tabs for 'Docs', 'Params' (which is selected), 'Authorization', 'Headers (7)', 'Body', 'Scripts', 'Tests', and 'Settings'. A 'Cookies' tab is also visible. Under the 'Params' tab, there is a table titled 'Query Params' with columns 'Key', 'Value', and 'Description'. One row is present with 'Key' as 'Key', 'Value' as 'Value', and 'Description' as 'Description'. In the bottom right corner of the interface, there is a status bar showing '200 OK', '8.46 s', '304 B', and other metrics.

HTTP <http://localhost:8080/api/ask/suggest products for vacation>

Save Share Send

GET <http://localhost:8080/api/ask/suggest products for vacation>

Docs Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	...

Body Cookies Headers (5) Test Results ⌚

200 OK 8.46 s 304 B ⚡ 🔍 🗃 ⚡

Raw Preview Visualize

```
1 For vacation, consider the "Waterproof Travel Backpack" for its multi-compartment design, padded laptop sleeve, and water-resistant fabric.
```

The screenshot shows the Postman application interface. At the top, the URL `http://localhost:8080/api/ask/suggest products for workout` is entered. The method is set to `GET`. On the right side, there are buttons for `Save`, `Share`, and a copy icon. Below the URL, the status bar shows `200 OK`, `4.06 s`, `400 B`, and a globe icon.

The main workspace displays the following details:

- Query Params:** A table with columns `Key`, `Value`, and `Description`. It contains one row with the value "Key".
- Body:** A table with columns `Raw`, `Preview`, and `Visualize`. The `Raw` tab is selected, showing the response body:

```
1 I suggest the "Yoga Mat (6mm, Non-Slip)" for your workout needs. It is a high-density, eco-friendly mat suitable for yoga, Pilates, or floor exercises. It features an anti-tear design, is lightweight, easy to roll, and sweat-resistant.
```
- Headers:** A table with 5 rows, labeled `Content-Type`, `Content-Length`, `Content-Encoding`, `Connection`, and `Date`.
- Test Results:** A table with 3 rows, labeled `Time`, `Memory`, and `Logs`.

```
// with chat memory advisors  
// GET http://localhost:8080/api/ask/suggest products for cooking
```

GET <http://localhost:8080/api/ask/suggest> products for cooking Send ▾

Docs Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results | ⌚

200 OK • 6.87 s • 670 B • 🌐 🕒

Raw ▾ Preview Visualize | ⟳ 🔍 ☰ ✖ 🖨️

```
1 Based on the provided context, here are some products you might find useful for cooking:  
2  
3 1. **Silicone Baking Mats (Set of 2)**  
4   - Description: Non-stick mats for baking with reusable, heat-resistant silicone.  
5   - Price: $13.00  
6  
7 2. **Kitchen Scale**  
8   - Description: Precise measurement for cooking and baking, with tare function.  
9   - Price: $14.75  
10  - Features: LCD screen, Slim design, Units in oz/g/ml/lb, Auto-off  
11  
12 If you need more suggestions or specific types of products, please let me know!
```

// After implementing PromptTemplate

The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/ask/suggest products for cooking
- Headers:** (7 items listed)
- Body:** (Empty)
- Tests:** (Empty)
- Scripts:** (Empty)
- Settings:** (Empty)
- Cookies:** (Empty)

The screenshot shows the Postman interface after sending the request, displaying the following results:

- Status:** 200 OK
- Time:** 2.27 s
- Size:** 326 B
- Headers:** (5 items listed)
- Raw Response:**

```
1 Title: "Silicone Baking Mats (Set of 2)"
2 Price: $13
3 Category: Cooking/Baking
4 Description: Non-stick mats for baking with reusable, heat-resistant silicone.
```
- Visualize:** (Empty)