Knowledge Base to Retell Webhook Integration

# 1. Extracting Chunks from DOCX

The DOCX file contains structured and semi-structured information such as daily schedules, menus, and other general instructions. To integrate this with an AI-driven assistant, we first parse the DOCX content into manageable 'chunks' of text. Each chunk is restricted to a token limit (e.g., 800 tokens) to be suitable for prompt input. Overlap between chunks ensures continuity.  
  
Key Steps:  
- Read DOCX content  
- Tokenize using an efficient tokenizer (e.g., from Hugging Face)  
- Create overlapping chunks  
- Save as a JSON for quick access

# 2. JSON Knowledge Base

The extracted chunks are stored in a JSON file (`kb\_store.json`). Each item contains:  
- city  
- location  
- category  
- chunk  
  
This allows fast lookup and filtering based on API requests.

# 3. Query Endpoint with FastAPI

A FastAPI server is created to serve queries to the knowledge base. The `/kb/query` endpoint accepts parameters like:  
- city  
- query  
- location (optional)  
  
It searches for the best matching chunk using both exact and fuzzy string matching to handle variations in input.

# 4. Fuzzy Matching for Robust Queries

To enhance query reliability, fuzzy string matching (e.g., using `fuzzywuzzy`) is used to compare user input with the knowledge base content.  
This improves handling of typos and variations in phrasing.

# 5. Webhook Integration with Retell.ai

Retell.ai supports integration with external knowledge bases through webhooks.  
  
Setup involves:  
- Hosting the FastAPI service (e.g., on Render or Replit)  
- Creating a public URL  
- Pointing Retell webhook configuration to `/kb/query`  
  
When the assistant needs to fetch knowledge, it triggers the webhook with parameters like city, location, and question. The FastAPI responds with a relevant chunk from the JSON knowledge base.

# 6. Deployment

To deploy the FastAPI server:  
- Use Render with the start command: `uvicorn retell\_webhook:app --host 0.0.0.0 --port $PORT`  
- Or run locally using: `uvicorn retell\_webhook:app --host 0.0.0.0 --port 8000`  
  
Ensure the API is publicly accessible so that Retell can reach it.