**HTTP REST Service for Text Storage and Summarization**

**Introduction**

This documentation outlines the design and implementation of an HTTP REST service using FastAPI for storing and retrieving a larger English text and generating a summary of that text. The service utilizes spaCy for natural language processing and summarization.

**Technologies Used**

- FastAPI: Web framework for building APIs with Python.

- spaCy: Natural language processing library for text processing.

- uvicorn: ASGI server for running the FastAPI application.

- Docker: Containerization platform for packaging and deploying the application.

**Endpoints**

1. Store Text

- Method: POST

- Endpoint: `/store\_text/`

- Request Body: `{"text": "Your text goes here"}`

- Description: Stores the provided text in memory.

2. Retrieve Text

- Method: GET

- Endpoint: `/retrieve\_text/`

- Description: Retrieves the stored text.

3. Retrieve Summary

- Method: GET

- Endpoint: `/retrieve\_summary/`

- Description: Generates and retrieves a summary of the stored text.

- Summary Generation: Uses spaCy to tokenize the text, calculate word frequencies, and identify the most important sentences based on word frequencies. The summary consists of 30% of the most important sentences.

**Implementation Details**

- Text Storage: The text is stored in memory as a global variable `stored\_text`.

- Text Retrieval: The stored text can be retrieved using the `/retrieve\_text/` endpoint.

- Summary Generation: The text is tokenized using spaCy, and word frequencies are calculated to determine the importance of each word. Sentences are scored based on the sum of word frequencies, and the top 30% of sentences are selected to form the summary.

- Error Handling: If no text is stored, a 404 HTTP exception is raised.

**Conclusion**

This HTTP REST service provides a simple yet effective way to store and retrieve a larger English text and generate a summary of that text. It leverages FastAPI for building the API, spaCy for natural language processing, and Docker for containerization, making it easy to deploy and use in various environments.