HSC Chatbot API Documentation

July 26, 2025

Base URL

```
http://your-server-ip:8080/
For local testing:
http://127.0.0.1:8080/
```

Endpoints

GET /

Description:
Renders the chat HTML page. This is the frontend interface for interacting with the chatbot.
Response:
Returns an HTML page.

POST /get

Description:
Accepts user input from a form and returns a response generated via a Retrieval-Augmented Generation (RAG) pipeline.
Request Parameters:

- `msg` (string): The user's question/input query.

Example Request (form-data or x-www-form-urlencoded):

```
POST /get
msg=Who is nelson mandela?
```

Backend Workflow

1. **PDF Loader:**
    - Loads and splits content using `RecursiveCharacterTextSplitter`
    - Source: `/Data` folder

2. **Embeddings:**
    - Model: `sentence-transformers/all-MiniLM-L6-v2`
3. **Vector Store:**
    - Qdrant Cloud (hsc_bot_qdrant2)
    - Similarity search (k=5)
4. **Language Model:**
    - `Qwen/Qwen3.0.6B` via Hugging Face Transformers

HSC Chatbot API Documentation (Continued)

- Integrated with LangChain using `HuggingFacePipeline`

Setup & Run

To start the server:

```
python app.py
```

Ensure:

- PDF files exist in `/Data/`
- Qdrant (if used) is loaded properly
- The model and dependencies are installed
- Port 8080 is open (if testing remotely)

Deprecation Warnings

Update imports:

```
from langchain_community.vectorstores import Qdrant
from langchain_community.embeddings import HuggingFaceEmbeddings
from langchain_community.document_loaders import PyPDFLoader,
   DirectoryLoader
```

Security Notes

- Do not expose your Qdrant API key in public code or URLs.
- Use environment variables and `.env` files.
- The current server runs in development mode - not safe for production.