# Llama Cloud -
# RAG implementation using Llama Parsing with Trulens Evaluation

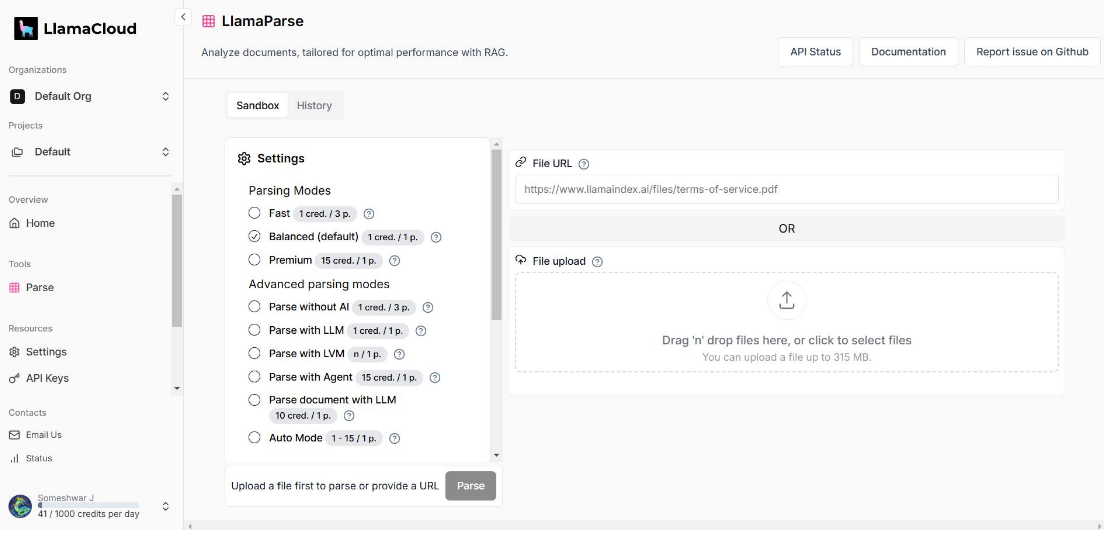
This project implements a simple Retrieval-Augmented Generation (RAG) pipeline. It involves parsing an uploaded document, setting up a RAG system, and evaluating its performance using TruLens.

## 1. Parsing the Uploaded Document

The first step in the pipeline is document parsing. The uploaded document is processed to extract text content, which is then used in the RAG pipeline. The parsing mechanism ensures the extracted content is clean and structured for retrieval.

## Steps Involved:

- Uploading the Document - The user provides a document (e.g., PDF, TXT).
- Extracting Text - The document is parsed to extract textual content for further processing.
- Preprocessing - Cleaning and structuring the text to ensure quality input for retrieval.



Settings

Parsing Modes

○ Fast 1 cred. / 3 p.
☑ Balanced (default) 1 cred. / 1 p.
○ Premium 15 cred. / 1 p.

Advanced parsing modes

○ Parse without AI 1 cred. / 3 p.
○ Parse with LLM 1 cred. / 1 p.
○ Parse with LVM n / 1 p.
○ Parse with Agent 15 cred. / 1 p.
○ Parse document with LLM 10 cred. / 1 p.
○ Auto Mode 1 - 15 / 1 p.

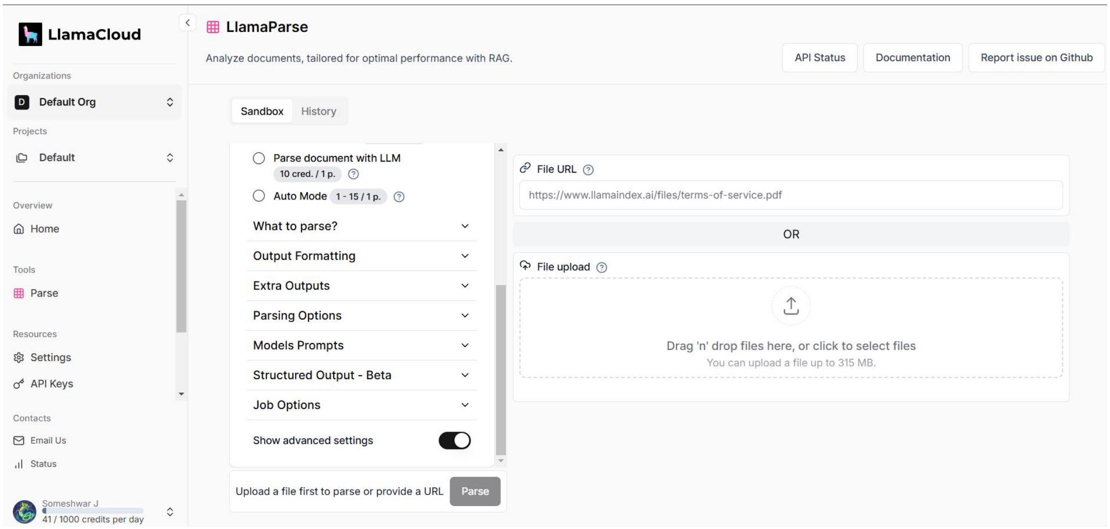
Upload a file first to parse or provide a URL

File URL: https://www.llamaindex.ai/files/terms-of-service.pdf

Parse
OR
File upload
Drag 'n' drop files here, or click to select files
You can upload a file up to 315 MB.

## Llama Cloud - Document Parsing and Agent Builder using Ollama

- Can parse, extract, and index a million of your most complex PDFs, Powerpoints, and more.
- Build Agent workflows on top of this.
- Link to create explore: Llama Cloud
- Documentation Link: LlamaParse Documentation



LlamaParse with Advanced Options

## 2. Building the RAG Application

Retrieval-Augmented Generation enhances language model responses by incorporating relevant retrieved documents. The RAG system consists of the following components:

### Components:

- Vector Database - Stores document embeddings for efficient retrieval.
- Embedding Model - Converts text into numerical representations.
- Retriever - Fetches the most relevant text passages based on queries.
- LLM (Large Language Model) - Generates responses using retrieved information.

### Workflow:

- Indexing the Document - The extracted text is converted into embeddings and stored in a vector database.
- Local Storage for Indexing - Since LlamaIndex is not free, the system uses local storage to save vectors and embeddings instead of a cloud-based solution.
- Retrieval - When a query is made, the retriever fetches relevant text chunks.
- Querying with LlamaIndex - The retriever utilizes the query capabilities from the LlamaIndex Python package to find the most relevant information.
- Generation - The retrieved text is passed to the LLM, which generates a final response.

### 3. Evaluating with TruLens

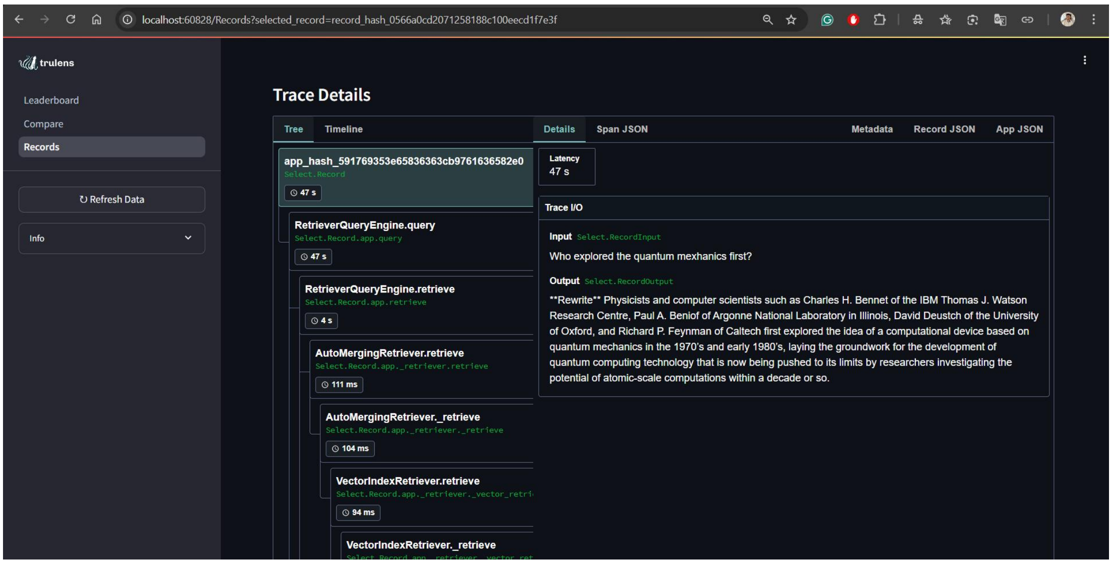
TruLens is used to assess the quality and reliability of the RAG system. It helps measure response relevance, faithfulness to retrieved data, and overall effectiveness.

**Evaluation Metrics:**

- Relevance - How well the retrieved text matches the query.
- Faithfulness - Whether the generated response accurately reflects the retrieved information.
- Latency - Measuring system response time.

**Steps:**

- Logging RAG Execution - Tracking retrieval and generation steps.
- Applying TruLens Metrics - Using TruLens to evaluate output quality.
- Interpreting Results - Analyzing performance and identifying areas for improvement.



Trace Details

app_hash_591769353e65836363cb9761636582e0
Select.Record
47 s

RetrieverQueryEngine.query
Select.Record.app.query
47 s

RetrieverQueryEngine.retrieve
Select.Record.app.retrieve
4 s

AutoMergingRetriever.retrieve
Select.Record.app._retriever.retrieve
111 ms

AutoMergingRetriever._retrieve
Select.Record.app._retriever._retrieve
104 ms

VectorIndexRetriever.retrieve
Select.Record.app._retriever._vector_retri
94 ms

VectorIndexRetriever._retrieve

Latency
47 s

Trace I/O

Input Select.RecordInput

Who explored the quantum mexhanics first?

Output Select.RecordOutput

**Rewrite** Physicists and computer scientists such as Charles H. Bennet of the IBM Thomas J. Watson Research Centre, Paul A. Beniof of Argonne National Laboratory in Illinois, David Deustch of the University of Oxford, and Richard P. Feynman of Caltech first explored the idea of a computational device based on quantum mechanics in the 1970's and early 1980's, laying the groundwork for the development of quantum computing technology that is now being pushed to its limits by researchers investigating the potential of atomic-scale computations within a decade or so.

Trulens UI

✧ Click here to explore the notebook