Building Enterprise RAG with RAGFlow

- 2 Days Course
- Hands-on and Lecture

Course Overview

Students stand up, configure, extend, and evaluate a multi-tenant RAGFlow deployment. They integrate custom chunking, hybrid search parameters, external APIs/MCP, and programmatically interface with RAGFlow via its HTTP/Python APIs to establish the enterprise platform boundary.

Who Should Attend

- Enterprise AI Engineers and RAG System Developers (Suggested)
- Solutions Architects and Platform Engineers (Suggested)
- Technical Leads integrating modular/packaged RAG platforms (Suggested)

What You'll Learn

- Understand RAGFlow core architecture, including Task Executors, DeepDoc, Vector Store adapters, and Model Connectors.
- Ingest and parse complex documents using DeepDoc and template-based chunking strategies.
- Configure and tune hybrid retrieval pipelines using BM25, dense-vector search, and cross-encoder reranking models.
- Understand RAGFlow's native security model and design patterns for integrating enterprise identity (OIDC/OAuth2) and authorization.
- Build visual agent workflows with RAGFlow, integrating external web crawlers, Model Context Protocol (MCP) servers, and custom HTTP tools.
- Programmatically interface with RAGFlow using the REST API and Python SDK, injecting dynamic runtime metadata filters and capturing structured responses with citations.

Outline

1. RAGFlow Architecture & Platform Core

- Understanding RAGFlow's core architecture: Task Executors, DeepDoc, Vector Store adapters, and Model Connectors
- Decoupling infrastructure: Connecting RAGFlow to external model servers (vLLM/TEI) and vector databases (OpenSearch/Infinity)
- Multi-tenancy, workspace separation, and dataset scoping within RAGFlow

2. DeepDoc Ingestion, Layout Analysis & Document Parsing

- Sourcing parser capabilities within RAGFlow: How DeepDoc handles multi-column layouts, nested tables, and OCR
- Document-centric chunking strategies: Naive, Book, Paper, Laws, Presentation, Picture, and Q&A templates

- Managing metadata, custom fields, and document status lifecycles inside RAGFlow Knowledge Bases

3. Hybrid Retrieval Configuration & Tuning

- Configure RAGFlow's hybrid retrieval pipeline using BM25 and dense-vector similarity
- Integrating cross-encoder reranking models within RAGFlow retrieval pipelines
- Visualizing retrieval hits, chunk scoring, and vector/lexical relevance weights
- Lab: RAGFlow Deployment, DeepDoc Ingestion & Retrieval Tuning
- Lab: Deploy RAGFlow using containerized orchestration connected to external LLM/embedding endpoints
- Lab: Ingest complex financial and clinical PDFs using DeepDoc across various chunking templates
- Lab: Compare candidate retrieval results while tuning vector/keyword weighting, similarity thresholds, top-k, and reranking

4. Enterprise Identity, Authorization & the RAGFlow Boundary

- Understanding RAGFlow's native security model: users, authentication, tenant/workspace permissions, dataset ownership, and knowledge-base sharing
- Integrating enterprise identity using OIDC/OAuth2 and mapping authenticated users into RAGFlow
- Separating authentication, platform permissions, dataset sharing, and retrieval authorization
- Using dataset boundaries and metadata filtering to enforce application-defined retrieval scope
- Extending RAGFlow with an application or API layer that translates enterprise roles and policy into permitted datasets and retrieval filters
- Authorization design patterns: native RAGFlow permissions, dataset-per-security-boundary, metadata-filtered retrieval, and application-controlled retrieval
- Understanding the limitations of each approach: administrative complexity, role/filter synchronization, filter correctness, policy drift, and large numbers of security boundaries
- Identifying when native RAGFlow controls are sufficient, when application-level augmentation is appropriate, and when authorization requirements justify a more custom architecture

5. Agent Workflows & Tool Integration

- Building visual workflow agents in RAGFlow: Knowledge Retrieval, Logic Nodes, and Categorization
- Extending agent capabilities: Connecting external web crawlers, Model Context Protocol (MCP) servers, and custom HTTP tools
- Balancing visual orchestration with application-owned business logic

6. Application Integration, Operations & APIs

- Interfacing with RAGFlow programmatically: Utilizing the REST API and Python SDK
- Injecting dynamic query filters and user context into API requests
- Session management, streaming responses, citation tracking, and telemetry monitoring
- Lab: RAGFlow Agents, Custom Tools & API Integration
- Lab: Build an multi-step agent inside RAGFlow that combines Knowledge Base retrieval with an external API tool
- Lab: Integrate RAGFlow programmatically into an external application using the Python SDK / REST API
- Lab: Inject dynamic runtime metadata filters via API calls and capture structured responses with source citations

Prerequisites

Next Courses