



ICT Training Center

Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

Corso avanzato per Venis S.p.A, Venezia, Italia

Novembre 2025

RETRIEVAL AUGMENTED GENERATION

PARTE 2

- ➡ *Chatbot* Ollama-HR-Venis-ITA e Gemini-HR-Venis-ENG
- ⚠ Approccio classico (informazione testuale estratta tramite documento)
 - 1 Modifica dipendenze di progetto
 - 2 Modifica configurazione e profilo applicativo
 - 3 Creazione *prompt templates* per strategia RAG
 - 4 Creazione *component* per popolamento *vector store* (dati Venis-HR)
 - 5 Test delle funzionalità con Postman/Insomnia

Dipendenze di sistema aggiuntive

```
...  
<dependency>  
  <groupId>org.springframework.ai</groupId>  
  <artifactId>spring-ai-tika-document-reader</artifactId>  
</dependency>  
...
```

Configurazione applicativo

```
spring:
  ...
  profiles:
    active: rag-document-to-vector-store
  autoconfigure:
    ...
```

File application-rag-document-to-vector-store.yml

```
demo:
  rag:
    prompt:
      system:
        ita: classpath:templates/get-rag-hr-data-system-ita-prompt.st
        eng: classpath:templates/get-rag-hr-data-system-eng-prompt.st
      user:
        ita:
        eng:
    vectorstore:
      qdrant:
        collection-name:
          ollama: vector_store_dtvts_ollama
          gemini: vector_store_dtvts_gemini
```

File templates/get-rag-hr-data-system-ita-prompt.st

Sei un assistente utile che risponde a domande sulle politiche di risorse umane dell'azienda Venis SpA ai suoi dipendenti.
Dato il contesto nella sezione DOCUMENTI e senza conoscenze pregresse, rispondi alla domanda del dipendente.
Se la risposta non si trova nella sezione DOCUMENTI, rispondi che non puoi rispondere alla domanda.

DOCUMENTI: <documenti>

File templates/get-rag-hr-data-system-eng-prompt.st

You are an helpful assistant, answering questions about Venis SpA policies to it's employees.
Given the context in the DOCUMENTS section and no prior knowledge, answer the employee's question.
If the answer is not in the DOCUMENTS section, then reply that you cannot answer to the question.

DOCUMENTS: <documenti>

Componente popolamento Qdrant - I

```
package it.venis.ai.spring.demo.rag;

...

@Component
@Profile("rag-document-to-vector-store")
public class DocumentDataLoader {

    private final VectorStore geminiVectorStore;
    private final VectorStore ollamaVectorStore;

    public DocumentDataLoader(@Qualifier("geminiVectorStore") VectorStore geminiVectorStore,
                               @Qualifier("ollamaVectorStore") VectorStore ollamaVectorStore) {
        this.geminiVectorStore = geminiVectorStore;
        this.ollamaVectorStore = ollamaVectorStore;
    }

    @Value("classpath:Venis_HR_Policies_ENG.pdf")
    Resource venisHREngDocument;

    @Value("classpath:Venis_Politiche_HR_ITA.pdf")
    Resource venisHRItaDocument;

    ...
}
```

Componente popolamento Qdrant - II

```
...
@PostConstruct
public void loadVenisHREngDocumentIntoVectorStore() {

    SearchRequest searchRequest = SearchRequest.builder()
        .query("Check")
        .similarityThresholdAll()
        .build();

    List<Document> similarDocs = geminiVectorStore.similaritySearch(searchRequest);

    if (similarDocs.size() == 0) {
        TikaDocumentReader tikaDocumentReader = new TikaDocumentReader(venisHREngDocument);

        List<Document> docs = tikaDocumentReader.get();

        TextSplitter textSplitter = TokenTextSplitter.builder()
            .withChunkSize(100)
            .withMaxNumChunks(400)
            .withKeepSeparator(true)
            .build();

        this.geminiVectorStore.add(textSplitter.split(docs));
    }
}

...
```

Componente popolamento Qdrant - III

```
...
@PostConstruct
public void loadVenisHRItaDocumentIntoVectorStore() {

    SearchRequest searchRequest = SearchRequest.builder()
        .query("Check")
        .similarityThresholdAll()
        .build();

    List<Document> similarDocs = ollamaVectorStore.similaritySearch(searchRequest);

    if (similarDocs.size() == 0) {
        TikaDocumentReader tikaDocumentReader = new TikaDocumentReader(venisHRItaDocument);

        List<Document> docs = tikaDocumentReader.get();

        TextSplitter textSplitter = TokenTextSplitter.builder()
            .withChunkSize(100)
            .withMaxNumChunks(400)
            .withKeepSeparator(false)
            .build();

        this.ollamaVectorStore.add(textSplitter.split(docs));
    }
}
```

⚠ Verificare la *dashboard* Qdrant (<http://172.17.0.1:6333/dashboard>)

The screenshot shows the Qdrant dashboard interface. At the top, there's a header with the Qdrant logo and navigation icons. The main section is titled 'Collections' and features a search bar labeled 'Search Collection'. Below the search bar is a table listing the collections. The table has columns for Name, Status, Points (Approx), Segments, Shards, Vectors Configuration (Name, Size, Distance), and Actions. Two collections are listed: 'vector_store_dtvvs_gemini' and 'vector_store_dtvvs_ollama', both with a status of 'green'.

Name	Status	Points (Approx)	Segments	Shards	Vectors Configuration (Name, Size, Distance)	Actions
vector_store_dtvvs_gemini	green	16	8	1	default 3072 Cosine	
vector_store_dtvvs_ollama	green	23	8	1	default 768 Cosine	

<https://github.com/simonescannapieco/spring-ai-advanced-dgroove-venis-code.git>

Branch:

8-spring-ai-gemini-ollama-rag-document-to-vector-store