

ICT Training Center







SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Novembre 2025



PARTE 2 - APPROCCIO DOCUMENT TO VECTOR STORE

PROGETTO SPRING AI APPLICAZIONE E PASSAGGI



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

PROGETTO SPRING AI



Dipendenze di sistema aggiuntive

PROGETTO SPRING AI



Configurazione applicativo

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

PROGETTO SPRING AI RAG - PROFILO APPLICATIVO



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_dtvs_clama
        gemini: vector_store_dtvs_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 \mbox{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 ${\tt DOCUMENTI}$, 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>

PROGETTO SPRING AI



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 venisHRTtaDocument:
```





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));
```



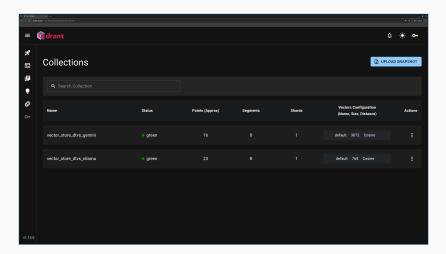
Componente popolamento Qdrant - III

```
@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));
```

AMBIENTE DI SVILUPPO POPOLAMENTO VECTOR STORE



▲ Verificare la dashboard Qdrant (http://172.17.0.1:6333/dashboard)



CODICE BRANCH DI RIFERIMENTO



 $\label{lem:https://github.com/simonescannapieco/spring-ai-advanced-dgroove-venis-code.git $Branch:$$

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