



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 3 - APPROCCIO RAG ADVISOR

- ➔ *Chatbot Ollama-HR-Venis-ITA classico e Gemini-HR-Venis-ENG con advisor*
- ⚠ *Approccio avanzato (informazione testuale estratta tramite advisor)*
 - 1 *Modifica file di configurazione RAG*
 - 2 *Modifica implementazione servizio*
 - 5 *Test delle funzionalità con Postman/Insomnia*

Configurazione Vector Store

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

...

@Configuration
public class RAGConfig {

    ...

    @Bean
    RetrievalAugmentationAdvisor geminiRetrievalAugmentationAdvisor(
        @Qualifier("geminiVectorStore") VectorStore geminiVectorStore) {

        return RetrievalAugmentationAdvisor.builder()
            .documentRetriever(
                VectorStoreDocumentRetriever.builder()
                    .vectorStore(geminiVectorStore)
                    .topK(4)
                    .similarityThreshold(.2)
                    .build()
            )
            .build();
    }
}
```

Implementazione servizio - I

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

...

@Service
@Configuration
public class RAGServiceImpl implements RAGService {

    ...

    private RetrievalAugmentationAdvisor geminiRetrievalAugmentationAdvisor;

    public RAGServiceImpl(
        @Qualifier("geminiChatClient") ChatClient geminiChatClient,
        @Qualifier("ollamaChatClient") ChatClient ollamaChatClient,
        @Qualifier("ollamaMemoryChatClient") ChatClient ollamaMemoryChatClient,
        @Qualifier("geminiVectorStore") VectorStore geminiVectorStore,
        @Qualifier("ollamaVectorStore") VectorStore ollamaVectorStore,
        @Qualifier("geminiRetrievalAugmentationAdvisor") RetrievalAugmentationAdvisor geminiRetrievalAugmentationAdvisor) {

        this.geminiChatClient = geminiChatClient;
        this.ollamaChatClient = ollamaChatClient;
        this.ollamaMemoryChatClient = ollamaMemoryChatClient;
        this.geminiVectorStore = geminiVectorStore;
        this.ollamaVectorStore = ollamaVectorStore;
        this.geminiRetrievalAugmentationAdvisor = geminiRetrievalAugmentationAdvisor;
    }

    @Value("${demo.rag.prompt.system.eng}")
    private Resource ragDataSystemEngPrompt;

    ...
}
```

Implementazione servizio - II

```
...
@Override
public Answer getGeminiRAGAnswer(QuestionRequest request) {
    /*
     * This code is no longer needed, because all is handled by the logic behind the
     * geminiRetrievalAugmentationAdvisor!
     */
    SearchRequest searchRequest = SearchRequest.builder()
        .query(request.body().question())
        .topK(4)
        .similarityThreshold(.2)
        .build();
    List<Document> similarDocs =
        geminiVectorStore.similaritySearch(searchRequest);
    String similarDocsString = similarDocs.stream()
        .map(Document::getText)
        .collect(Collectors.joining(System.lineSeparator()));
    /*
    ...

```

Implementazione servizio - III

```
...
return new Answer(this.geminiChatClient.prompt()
    .advisors(List.of(new SimpleLoggerAdvisor(), geminiRetrievalAugmentationAdvisor))
    /*
     * The geminiRetrievalAugmentationAdvisor takes also care to create
     * a dedicated system prompt to handle the RAG strategy!
     */
    .system(s -> s.text(this.ragDataSystemEngPrompt)
    .params(Map.of("documenti", similarDocsString)))
    /*
    .user(request.body().question())
    /*
     * The template rendered is now useless, since the system prompt is
     * automatically created!
     */
    .templateRenderer(StTemplateRenderer.builder().startDelimiterToken('<')
    .endDelimiterToken('>')
    .build())
    /*
    .call()
    .content());
}
...
}
```


<https://github.com/simonescannapieco/spring-ai-advanced-dgroove-venis-code.git>
Branch: 9-spring-ai-gemini-ollama-rag-advisor