



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 1

 Storico della conversazione valida per singola sessione

- ➔ Configurazione *JDBC H2 chat memory* per ChatClient Ollama
 - 1 Modifica al `pom.xml` per dipendenze Spring AI JDBC, H2 e *devtools*
 - 2 Modifica ad `application.yml`
 - 3 Creazione schema H2
 - 4 Modifica ai bean ChatClient per Ollama
 - 5 Test delle funzionalità con Postman/Insomnia

AMBIENTE DI SVILUPPO

CREAZIONE ACCOUNT HUGGINGFACE

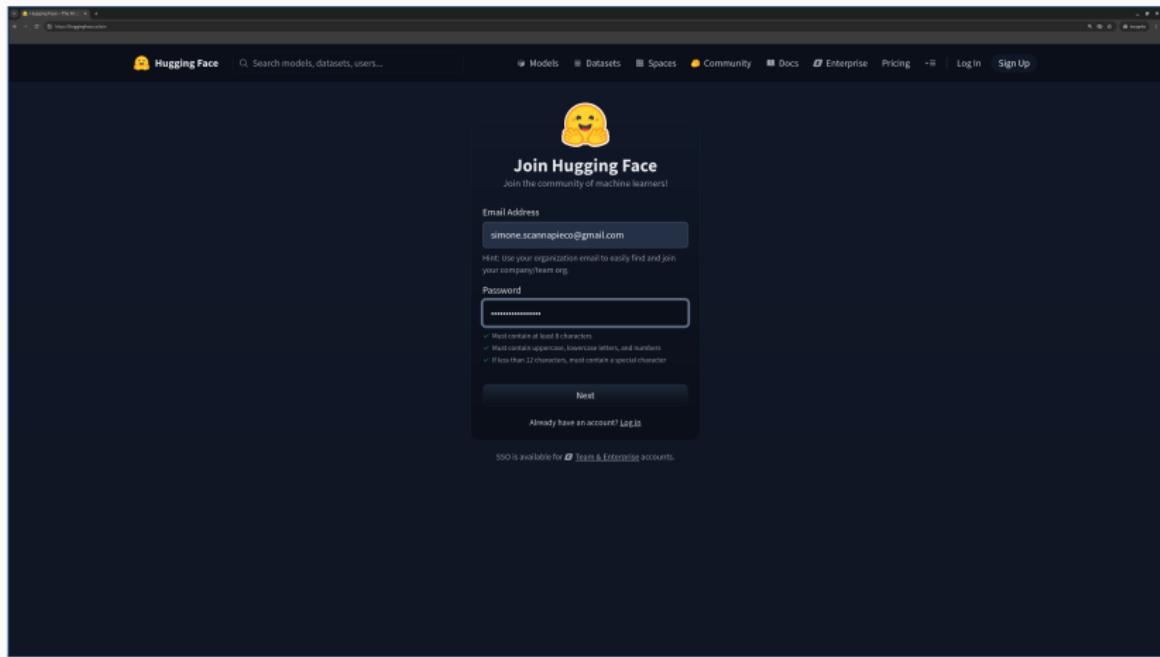
- 1 Accedere al portale <https://huggingface.co/>

The screenshot shows the Hugging Face website. At the top, there is a navigation bar with links for Models, Datasets, Spaces, Community, Docs, Enterprise, Pricing, Log In, and Sign Up. The Sign Up button is highlighted with a red box. Below the navigation bar, there is a search bar and a sidebar with categories like Multimodal, Computer Vision, NLP, and Tabular. The main content area features a large emoji of a smiling face and the text "The AI community building the future." Below this, it says "The platform where the machine learning community collaborates on models, datasets, and applications." There are two buttons: "Explore AI Apps" and "Browse 1M+ models". To the right, there is a list of trending models, each with a thumbnail, name, description, and metrics like "Test Generation" and "Updated 2 days ago". Some models shown include net-a-llama-2-760, stable-diffusion-xl-base-0.9, openai/llmchat, llyasaili/cetinNet-v1.1, pascalsense/mazoscope_v2_26, mta-llama-2-11b, tllium/Faiss-4B-struct, klausm/wikitext-2B-v2.0, ComfyUI/table-diffusion-v1.4, and stable-diffusion-v1.4. At the bottom, there is a "Trending on 🤖 this week" section.

AMBIENTE DI SVILUPPO

CREAZIONE ACCOUNT HUGGINGFACE

- Creare una nuova utenza premendo il pulsante Sign up



AMBIENTE DI SVILUPPO

CREAZIONE ACCOUNT HUGGINGFACE

3 Accedere al portale e selezionare Access tokens nel menu in alto a destra

The screenshot shows the Hugging Face website interface. On the left, there's a sidebar with options like 'Profile', 'Inbox (0)', 'Settings', 'Billing', and 'Get PRO'. Below that are sections for 'Organizations' and 'Resources'. The main area shows a 'Following' list with items from 'essential' (They might open source something soon), 'yogito' (Pioneering in AI video generation technology), and 'Qwen' (Developing advanced large language and vision models). On the right, there's a 'Trending' section and a 'Create organization' button. At the very bottom, there's a 'Wav2.24B Fast' card. The top navigation bar includes 'Models', 'Datasets', 'Spaces', 'Community', 'Docs', 'Enterprise', and 'Pricing'. A blue circle with the number '3' is overlaid on the top left of the screenshot.

AMBIENTE DI SVILUPPO

CREAZIONE ACCOUNT HUGGINGFACE

4 Premere sul pulsante Create new token

The screenshot shows the Hugging Face Hub interface. On the left, there's a sidebar with various account management options like Profile, Account, Authentication, Organizations, Billing, and several sections under Access Tokens. The 'Access Tokens' section is currently selected. At the top right of this section, there's a button labeled '+ Create new token' which is highlighted with a red dashed box. Below this button, there's a note about access tokens and a warning to not share them. A table lists existing access tokens with columns for Name, Value, Last Refreshed Date, Last Used Date, and Permissions (with a 'WRITE' row shown). At the bottom of the page, there are links for Local Apps and Hardware, Gated Repositories, Content Preferences, Connected Apps, MCP, and Theme, along with an 'Upgrade to Pro' button.

Name	Value	Last Refreshed Date	Last Used Date	Permissions
SPRING-AI-ADVANCED-DGROOVE@VENIS	H_L_HFYX	about 1 hour ago	-	WRITE
ITS-LAST COOKIE-2024	H_L_cEp	13 days ago	Jan 9	READ
CCAI_AIPERUVIANUM	H_L_wHFU	Jul 23	-	READ
SENTENCE-TRANSFORMERS-TEST-2025	H_L_XRW	May 26	May 26	READ
ITS-LAST-SUSE-2024	H_L_mbQL	Dec 2, 2024	Dec 6, 2024	READ

5 Selezionare token di tipo Write, denominare il token e premere Create token

The screenshot shows the 'Create new Access Token' page in the Hugging Face web interface. On the left, there's a sidebar with various account management options like Profile, Account, Authentication, etc. The main form has 'Token type' set to 'Write' (highlighted with a red box). The 'Token name' field contains 'SPRING AI-ADVANCED-DGROOVE@VENIS'. Below the form, a note states: 'This token has read and write access to all your and your orgs resources and can make calls to Inference Providers on your behalf.' A large blue 'Create token' button is at the bottom of the form.

AMBIENTE DI SVILUPPO

CREAZIONE ACCOUNT HUGGINGFACE

6 Cercare il modello di *embedding*, leggendone la card

The screenshot shows the Hugging Face platform interface. At the top, there is a search bar with the placeholder "Simone Scannapieco". Below the search bar, the results for "Simone Scannapieco" are displayed. The first result is "Simone-Scannapieco/sentence-bert-base-italian-v2", which is described as "See 1 model results for 'Simone Scannapieco'".

The main content area features a large image of a smiling emoji. Below it, the text "The AI community building the future." is displayed in a large, bold font. A subtitle below reads: "The platform where the machine learning community collaborates on models, datasets, and applications." There are two buttons at the bottom left: "Explore AI Apps" and "Browse 1M+ models".

On the right side of the screen, there is a sidebar with various categories: Models, Datasets, Spaces, Community, Docs, Enterprise, Pricing, Log In, and Sign Up. Below these, there is a "Trending on 🤖 this week" section.

The main search results area has tabs for Libraries, Datasets, Languages, Licenses, and Other. It lists several models, each with a thumbnail, name, description, and a "View Model" button. Some models shown include:

- meta-llama/Llama-2-7B
- stabilityai/stable-diffusion-xl-base-0.9
- openchat/openchat
- llyasizel/Greyscale-v1.1
- carlsanne/zero-shot_v2_XXL
- meta-llama/Llama-2-13B
- flirtei/falcon-4B-176v1.1
- MirroredVisionCoCo-15B-V2.0
- CompVis/latent-diffusion-v1.4
- stabilityai/latent-diffusion-v1.4
- balefire/space-70-8k-v1-test

AMBIENTE DI SVILUPPO

CREAZIONE ACCOUNT HUGGINGFACE

6 Cercare il modello di *embedding*, leggendone la card

The screenshot shows the Hugging Face Model Card interface for the model `sentence-bert-base-italian-xxl-uncased-F32-GGUF`. The card includes the following details:

- Model Card:** Shows the model's name, last updated (16 days ago), and its use of the GGUF binary format.
- Community:** Includes links to Sentence Similarity, sentence-transformers, and other related models like `PhilipMay/stsb_muli_mt`.
- Model Card:** Provides a detailed description of the model as a `sentence-transformers` model mapping sentences and paragraphs to a 768-dimensional dense vector space for tasks like clustering or semantic search. It notes dependencies on `dbmdz/bert-base-italian-xxl-uncased` and `nickrocks/bert-base-italian-xxl-uncased`.
- Hardware compatibility:** Lists supported architectures: 32-bit and F32 (441.85).
- Inference Providers:** Shows support for Sentence Similarity.
- Base model:** Points to the `sickipedia/sentence-bert-base-italian-xxl-uncased` model.
- Dataset used to train:** References the `PhilipMay/stsb_muli_mt` dataset.

File launch.json

```
{  
    // Use IntelliSense to learn about possible attributes.  
    // Hover to view descriptions of existing attributes.  
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387  
    "version": "0.2.0",  
    "configurations": [  
        {  
            "type": "java",  
            "name": "Launch Current File",  
            "request": "launch",  
            "mainClass": "${file}"  
        },  
        {  
            "type": "java",  
            "name": "DemoApplication",  
            "request": "launch",  
            "mainClass": "it.venis.ai.spring.demo.DemoApplication",  
            "projectName": "demo",  
            "env": {  
                "GOOGLE_AI_API_KEY": "...",  
                "HUGGING_FACE_HUB_TOKEN": "..."  
            }  
        }  

```

File settings.json

```
{  
    "java.compile.nullAnalysis.mode": "disabled",  
    "java.configuration.updateBuildConfiguration": "interactive",  
    "java.test.config": {  
        "env": {  
            "GOOGLE_AI_API_KEY": "...",  
            "HUGGING_FACE_HUB_TOKEN": "..."  
        }  
    }  
}
```

File docker-compose.yml

```
services:  
  ...  
  spring-ai-vector-store:  
    image: qdrant/qdrant:${QDRANT_VERSION:-latest}  
    hostname: spring-ai-vector-store  
    container_name: spring_ai_vector_store  
    ports:  
      - ${QDRANT_HTTP_PORT:-6333}:6333  
      - ${QDRANT_GRPC_PORT:-6334}:6334  
    volumes:  
      - spring_ai_vector_store:/qdrant/storage  
    restart: unless-stopped  
  
volumes:  
  ...  
  spring_ai_vector_store:  
    name: spring_ai_vector_store
```

File spring-ai.env

```
...  
# qdrant configuration  
QDRANT_VERSION=v1.13.0  
QDRANT_HTTP_PORT=6333  
QDRANT_GRPC_PORT=6334  
  
# default: latest  
# default: 6333  
# default: 6334
```

Dipendenze di sistema aggiuntive

```
...
<dependency>
    <groupId>org.springframework.ai</groupId>
    <artifactId>spring-ai-rag</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.ai</groupId>
    <artifactId>spring-ai-advisors-vector-store</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.ai</groupId>
    <artifactId>spring-ai-starter-vector-store-qdrant</artifactId>
</dependency>
...
...
```

Configurazione applicativo

```
spring:
  ...
  autoconfigure:
    exclude:
      - org.springframework.ai.vectorstore.qdrant.autoconfigure.QdrantVectorStoreAutoConfiguration
        # We must disable Vector Store auto-configuration because of two different EmbeddingModel beans
        # (OpenAI-Gemini and Ollama). The goal is to have two different vector collections, one for each
        # family of LLM.
  ai:
    ollama:
      ...
      embedding:
        model: hf.co/Simone-Scannapieco/sentence-bert-base-italian-xxl-uncased-F32-GGUF
  openai:
    ...
    embedding:
      options:
        model: gemini-embedding-001
  vectorstore:
    qdrant:
      initialize-schema: true
      host: 172.17.0.1
      port: 6334
  ...
```

File erase_llm_volumes.sh

```
#!/bin/bash

volume_name=spring_ai_llm

docker volume rm $volume_name
```

File erase_vector_store_volumes.sh

```
#!/bin/bash

volume_name=spring_ai_vector_store

docker volume rm $volume_name
```

File get-rag-data-system-ita-prompt.st

Sei un assistente AI in grado di rispondere alle domande dell'utente solo in base al contesto fornito dalla sezione DOCUMENTI.

Se la risposta non è presente nella sezione DOCUMENTI, informa l'utente di non sapere la risposta.

DOCUMENTI: <documenti>

File get-rag-data-system-eng-prompt.st

You are a helpful assistant, answering questions based on the given context in the DOCUMENTS section and no prior knowledge.

If the answer is not in the DOCUMENTS section, then reply that you cannot answer to the question.

DOCUMENTS: <documenti>

Configurazione Vector Store - I

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

import org.springframework.ai.embedding.EmbeddingModel;
import org.springframework.ai.openai.OpenAiEmbeddingModel;
import org.springframework.ai.vectorstore.VectorStore;
import org.springframework.ai.vectorstore.qdrant.QdrantVectorStore;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import io.qdrant.client.QdrantClient;
import io.qdrant.client.QdrantGrpcClient;

@Configuration
public class RAGConfig {

    @Autowired
    public OpenAiEmbeddingModel openAiEmbeddingModel;

    @Autowired
    public EmbeddingModel ollamaEmbeddingModel;

    @Value("${spring.ai.vectorstore.qdrant.host:#{null}}")
    private String qdrantHost;
    @Value("${spring.ai.vectorstore.qdrant.port:#{null}}")
    private String qdrantPort;
    @Value("${spring.ai.vectorstore.qdrant.use-tls:#{null}}")
    private String useTls;

    ...
}
```

Configurazione Vector Store - II

```
...  
@Bean  
public QdrantClient qdrantClient() {  
    QdrantGrpcClient.Builder grpcClientBuilder = QdrantGrpcClient.newBuilder()  
        .qdrantHost == null ? "localhost" : qdrantHost,  
        .qdrantPort == null ? 6334 : Integer.valueOf(qdrantPort),  
        .useTls == null ? false : Boolean.valueOf(useTls);  
    return new QdrantClient(grpcClientBuilder.build());  
}  
  
@Value("${spring.ai.vectorstore.qdrant.collection-name.gemini:#{null}}")  
private String qdrantCollectionNameGemini;  
@Value("${spring.ai.vectorstore.qdrant.collection-name.ollama:#{null}}")  
private String qdrantCollectionNameOllama;  
@Value("${spring.ai.vectorstore.qdrant.initialize-schema:#{null}}")  
private String qdrantInitializeSchema;  
  
@Bean  
public VectorStore geminiVectorStore(QdrantClient qdrantClient,  
    @Qualifier("openAiEmbeddingModel") EmbeddingModel geminiEmbeddingModel) {  
    return QdrantVectorStore.builder(qdrantClient, geminiEmbeddingModel)  
        .collectionName(qdrantCollectionNameGemini == null ? "vector_store_gemini" : qdrantCollectionNameGemini)  
        .initializeSchema(qdrantInitializeSchema == null ? false : Boolean.valueOf(qdrantInitializeSchema))  
        .build();  
}  
  
@Bean  
public VectorStore ollamaVectorStore(QdrantClient qdrantClient,  
    @Qualifier("ollamaEmbeddingModel") EmbeddingModel ollamaEmbeddingModel) {  
    return QdrantVectorStore.builder(qdrantClient, ollamaEmbeddingModel)  
        .collectionName(qdrantCollectionNameOllama == null ? "vector_store_ollama" : qdrantCollectionNameOllama)  
        .initializeSchema(qdrantInitializeSchema == null ? false : Boolean.valueOf(qdrantInitializeSchema))  
        .build();  
}
```

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

The screenshot shows the Qdrant dashboard interface. At the top, there's a search bar labeled "Search Collection". Below it, a table lists two collections:

Name	Status	Points (Approx)	Segments	Shards	Vectors Configuration (Name, Size, Distance)	Actions
vector_store_gemini	green	18	8	1	default 3072 Cosine	⋮
vector_store_oilama	green	17	8	1	default 768 Cosine	⋮

At the bottom left, it says "v1.19.0".



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

Branch: 7-spring-ai-gemini-ollama-rag-text-to-vector-store