



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



Note			



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Settembre 2025

Note		

SPRING AI: INTRODUZIONE	
	_ _ _
	_
	SPRING AI: INTRODUZIONE

SPRING AI L'EVOLUZIONE DELL'ECOSISTEMA SPRING



- Framework per integrare l'Intelligenza Artificiale nelle applicazioni Spring Boot
- Astrazione unificata per diversi provider AI (OpenAI, Azure OpenAI, AWS Bedrock, Google Vertex AI)
- Integrazione nativa con l'ecosistema Spring esistente
- Supporto per Vector Stores, Embedding e Retrieval-Augmented Generation (RAG)

🐸 Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

Note		

SPRING AI PERCHÉ SPRING AI?



- Problema: ogni provider AI ha API e modelli diversi
- Soluzione Spring AI: interfaccia unificata e portabile
- Principi Spring:
 - Dependency Injection
 - Auto-configuration
 - Testabilità
 - Observability
- Integrazione con Spring Boot, Spring Security, Spring Data

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

Note	

	ARCHITETTURA DI SPRING AI
Note	

Γ



Note			

CHAT CLIENT IL CUORE DELL'INTERAZIONE AI



- ChatClient: interfaccia principale per conversazioni Al
- ♦ Fluent API per costruire richieste complesse
- Supporto per streaming, function calling e multimodal

Note

VECTOR STORES GESTIONE DELLA MEMORIA SEMANTICA



- Vector Store: database ottimizzati per similarity search
- Supporto per Redis, Chroma, Weaviate, Pinecone
- Integrazione con Spring Data patterns

EMBEDDING MODELS TRASFORMAZIONE TESTO IN VETTORI



- **EmbeddingClient**: interfaccia per modelli di embedding
- Trasforma testo in rappresentazioni vettoriali per similarity search
- Supporto per OpenAl, Ollama, Azure OpenAl

```
OService
public class DocumentService {

private final EmbeddingClient embeddingClient;
private final VectorStore vectorStore;

public void addDocument(String content, Map<String, Object> metadata) {

// Crea embedding del contenuto

List<Float> embedding = embeddingClient.embed(content);

// Salva nel vector store

Document document = new Document(content, metadata);
document.setEmbedding(embedding);

Simone Scannapieco
vectorStore.add(List.of(document));

Wenis S.p.A, Venezia, IT

8/15

}
```

Note

VANTAGGI DI SPRING AI Note
Note
Note
Note

VANTAGGI DI SPRING AI PERCHÉ SCEGLIERE SPRING AI



- Portabilità tra provider: cambio di provider Al senza modificare codice business
- 1 Integrazione Spring nativa: sfrutta tutte le funzionalità dell'ecosistema
- Auto-configuration: configurazione automatica basata su dependencies
- Testabilità migliorata: mock e test integration semplificati
- Observability: metriche e monitoring integrati
- Gestione errori: retry policies e fallback automatici

Note		

PORTABILITÀ TRA PROVIDER Un codice, molti provider





Note

Il business logic rimane identico!

AUTO-CONFIGURATION ZERO BOILERPLATE CODE



- application.properties definisce tutto
- ©ConditionalOn* annotations per configurazione condizionale

```
# OpenAI Configuration
spring.ai.openai.api-key=${OPENAI_API_KEY}
spring.ai.openai.chat.model=gpt-4
spring.ai.openai.chat.temperature=0.7

# Vector Store Configuration
spring.ai.vectorstore.redis.uri=redis://localhost:6379
spring.ai.vectorstore.redis.index=spring-ai-index
spring.ai.vectorstore.redis.prefix=doc:

# Embedding Configuration
spring.ai.openai.embedding.model=text-embedding-ada-002
```

👺 Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

Note		





```
Simone
                 12/15
Note
```


	SVANTAGGI E LIMITAZIONI
Note	

Г

SVANTAGGI DI SPRING AI CONSIDERAZIONI E LIMITAZIONI



- Progetto giovane: API ancora in evoluzione (versione 1.x)
- ♦ Astrazione overhead: layer aggiuntivo rispetto alle API native
- Dipendenza dall'ecosistema Spring: lock-in tecnologico
- Supporto provider limitato: non tutti i provider sono supportati
- Funzionalità avanzate: alcuni provider offrono feature specifiche non astratte
- ♦ Learning curve: necessario apprendere nuovi concetti Al oltre Spring

Note	

API IN EVOLUZIONE RISCHI DEL PROGETTO GIOVANE



- Breaking changes: possibili modifiche incompatibili tra versioni
- Documentazione incompleta: esempi e best practices in sviluppo
- Community piccola: meno risorse e troubleshooting disponibili

Spring AI è attualmente in versione 1.0.1. Sebbene l'API principale sia stabile, alcune funzionalità potrebbero cambiare nelle future versioni minor.

Considerazione per progetti enterprise

🐸 Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

Note		

OVERHEAD DELL'ASTRAZIONE IL PREZZO DELLA PORTABILITÀ



```
// Chiamata diretta OpenAI (senza Spring AI)
OpenAiService service = new OpenAiService("sk-...");
CompletionRequest request = CompletionRequest.builder()
    .model("gpt-4")
    .prompt("Explain Spring Boot")
    .temperature(0.7)
    .maxTokens(150)
    .build();
CompletionResult result = service.createCompletion(request);

// VS Spring AI (più semplice ma con overhead)
ChatClient client = ChatClient.builder().build();
String response = client.prompt()
    .user("Explain Spring Boot")
    .call()
    .content();
```

Semplicità vs Performance: trade-off da valutare

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

Note		