



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

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

Note			

	SPRING AI INTRODUZIONE
Note	

# SPRING AI L'EVOLUZIONE DELL'ECOSISTEMA SPRING



- Ultimo progetto dell'ecosistema Spring
- Obiettivo: Integrare la GENAI nelle applicazioni Spring Boot
- Astrazione unificata per diversi provider Al
  - Anthropic (Claude)
  - OpenAl (GPT)
  - Microsoft (Azure OpenAI)
  - Soogle (Al Studio, Vertex Al)
- Integrazione nativa con l'ecosistema Spring esistente

👺 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: interfaccia unificata e portabile
- Aderenza ai principi Spring cardine
  - O IoC + DI (Inversion of Control and Dependency Injection)
  - AOP (Aspect Oriented Programming)
  - Astrazione
  - Onfigurazione dichiarativa e *in-code*
  - ₱ POJO
  - Modularità
- Integrazione con Spring Boot, Spring Security, Spring Data

Spring Al - Corso base

m Venis S.p.A, Venezia, IT

Note	

	VANTAGGI DI SPRING AI
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	9
------	---

#### 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	



```
class ChatServiceTest {
           private ChatClient chatClient;
           private ChatService chatService;
           void shouldGenerateResponse() {
                when(chatClient.prompt().user("Hello").call().content()
                        .thenReturn("Hello! How can I help you?");
                String response = chatService.chat("Hello");
                           Spring AI - Corso base
Simone Scannapieco
                                                      m Venis S.p.A, Venezia, IT
                                                                                8/11
                assertThat(response).isEqualTo("Hello! How can I help y u?");
Note
```

	SVANTAGGI E LIMITAZIONI	
Note		

## 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

Simone Scannapieco
Spring AI - Corso base

🏛 Venis S.p.A, Venezia, IT

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( )
    .maxTokens( )
    .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

Note		