



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

Settembre 2025

SPRING AI

INTRODUZIONE

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

- ➔ **Problema:** ogni *provider* AI ha API e modelli diversi
- ➔ **Soluzione:** interfaccia unificata e portabile
- ➔ Aderenza ai principi Spring cardine
 - ➔ IoC + DI (Inversion of Control and Dependency Injection)
 - ➔ AOP (Aspect Oriented Programming)
 - ➔ Astrazione
 - ➔ Configurazione dichiarativa e *in-code*
 - ➔ POJO
 - ➔ Modularità
- ➔ **Integrazione** con Spring Boot, Spring Security, Spring Data

VANTAGGI

- ➡ **Portabilità tra provider:** cambio di *provider* AI senza modificare codice
- ➡ **Integrazione Spring nativa:** sfrutta tutte le funzionalità dell'ecosistema
- ➡ **Auto-configuration:** configurazione automatica basata su *dependencies* (Gradle, Maven, ...)
- ➡ **Testabilità migliorata:** *mock* e *test integration* semplificati
- ➡ **Observability:** metriche e *monitoring* integrati
- ➡ **Gestione errori:** *retry policies* e *fallback* automatici

SVANTAGGI E LIMITAZIONI

- ➡ **Progetto giovane:** API ancora in evoluzione (versione 1.0.2)
- ➡ **Abstraction 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 AI oltre Spring

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