



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

CONTROLLI DI OUTPUT LLM

- 1 Modifica `Question.java` per integrazione UUID
- 2 Modifica `application.yml` con tutti possibili parametri
- 3 *Tweak* temperatura/Top-P/Top-K (Gemini vs Llama)
- 4 *Tweak stop sequence* per richiesta di liste
- 5 Test delle funzionalità con Postman/Insomnia

Modello per domanda

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

import java.util.UUID;

public record Question(UUID id, String question) {

    public Question(String question) {
        this(UUID.randomUUID(), question);
    }

}
```

- ⚠ Proprietà non utilizzabili (es. `top-k` per OpenAI)
- ⚠ Proprietà limitate a priori (es. `temperature` $\in [0, 2]$ per OpenAI)
- ⚠ Proprietà non stabili (es. `frequency-penalty`)
- ⚠ Proprietà mutuamente esclusive (es. `max-tokens` e `max-completion-tokens`)

File `application.yml`

```
spring:
  application:
    name: demo
  ai:
    openai:
      api-key: ${GOOGLE_AI_API_KEY}
      base-url: https://generativelanguage.googleapis.com/v1beta/openai
      chat:
        completions-path: /chat/completions
        options:
          model: gemini-2.0-flash-lite
          temperature: 2
          #top-p: .99
          #top-k: 30
          #max-tokens: 100
          #max-completion-tokens: 100
          #frequency-penalty: .2
          #presence-penalty: .2
          stop-sequences:
            - 3
```

<https://github.com/simonescannapieco/spring-ai-base-dgroove-venis-code.git>
Branch: 12-spring-ai-gemini-llm-output-controls