



This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# SPRING AI

## GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Novembre 2025

### Note

[illegible]



## **SETUP PROGETTO SPRING AI**

## MULTI-CONFIGURAZIONE

### Note

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

-  Simone Scannapieco

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

- 1 Creazione `docker-compose.yml` per servizio Docker Ollama
- 2 Creazione *file* variabili di ambiente per servizio Ollama
- 3 Creazione *script* per *start*, *stop* ed eliminazione servizi Docker
- 4 Creazione configurazione multi-LLM
- 5 Creazione modelli per domanda e risposta
- 6 Creazione interfaccia ed implementazione del servizio di richiesta
- 7 Creazione del controllore MVC
- 8 Test delle funzionalità con Postman/Insomnia

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page, typical of notebook or legal stationery. There are no margins, text, or other markings present.

```
services:
  spring-ai-llm-gpu:
    image: ollama/ollama:${OLLAMA_VERSION:-latest}
    hostname: spring-ai-llm
    container_name: spring_ai_llm
    environment:
      OLLAMA_HOST: "${OLLAMA_HOST:-0.0.0.0}:${OLLAMA_PORT:-11434}"
      OLLAMA_DEBUG: ${OLLAMA_DEBUG:-false}
      OLLAMA_FLASH_ATTENTION: ${OLLAMA_FLASH_ATTENTION:-false}
      OLLAMA_KEEP_ALIVE: ${OLLAMA_KEEP_ALIVE:-"5m"}
      OLLAMA_MAX_LOADED_MODELS: ${OLLAMA_MAX_LOADED_MODELS:-1}
      OLLAMA_NUM_PARALLEL: ${OLLAMA_NUM_PARALLEL:-1}
    expose:
      - ${OLLAMA_PORT:-11434}
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]
    volumes:
      - spring_ai_llm:/root/.ollama
    restart: unless-stopped
    profiles: [llm-gpu]
```

## Note

[illegible]

...

```
volumes:
  spring_ai_llm:
    name: spring_ai_llm
```

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

```
# MODE:
# "llm-cpu" --> Large Language Model in cpu mode
# "llm-gpu" --> Large Language Model in gpu mode
MODE=llm-cpu
COMPOSE_PROFILES=${MODE}
LOG_LEVEL=WARNING                                     # default: WARNING

# Ollama configuration
#OLLAMA_VERSION=0.1.39                                # default: latest
OLLAMA_HOST=spring-ai-llm                             # default: 0.0.0.0
OLLAMA_PORT=11434                                       # default: 11434
OLLAMA_DEBUG=false                                     # default: false
OLLAMA_FLASH_ATTENTION=false                           # default: false
OLLAMA_KEEP_ALIVE="5m"                                 # default: "5m"
OLLAMA_MAX_LOADED_MODELS=2                             # default: 1
OLLAMA_NUM_PARALLEL=1                                  # default: 1
```

### Note

[illegible]



## **File** start\_spring\_ai\_services.sh

```
#!/bin/bash

stack=spring-ai-demo-services
rmi=local
file=docker-compose.yml
envfile=spring-ai.env

if [ -z ${1+x} ];
then rmi=local;
else rmi=$1;
fi

## --rmi flag must be one of: all, local
docker compose -f $file -p $stack --env-file $envfile up --build --remove-orphans --force-recreate --detach
```

### Note

[illegible]

```
#!/bin/bash

stack=spring-ai-demo-services
rmi=local
file=docker-compose.yml
envfile=spring-ai.env

if [ -z ${1+x} ];
then rmi=local;
else rmi=$1;
fi

## --rmi flag must be one of: all, local
docker compose -f $file -p $stack --env-file $envfile down --rmi $rmi
```

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```
#!/bin/bash

stack=spring-ai-demo
rmi=local
file=docker-compose.yml
envfile=spring-ai.env

if [ -z ${1+x} ];
then rmi=local;
else rmi=$1;
fi

## --rmi flag must be one of: all, local
docker compose -f $file -p $stack --env-file $envfile down --rmi $rmi --volumes
```

[illegible]

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

import org.springframework.ai.chat.client.ChatClient;
import org.springframework.ai.ollama.OllamaChatModel;
import org.springframework.ai.openai.OpenAiChatModel;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Primary;

@Configuration
public class ChatClientConfig {

    @Bean
    @Primary
    public ChatClient openAiChatClient(OpenAiChatModel openaiChatModel) {
        return ChatClient.create(openaiChatModel);
        /*
         * or:
         * ChatClient.Builder chatClientBuilder = ChatClient.builder(openaiChatModel);
         * return chatClientBuilder.build();
         */
    }

    @Bean
    public ChatClient ollamaChatClient(OllamaChatModel ollamaChatModel) {
        ChatClient.Builder chatClientBuilder = ChatClient.builder(ollamaChatModel);
        return chatClientBuilder.build();
        /*
         * or:
         * return ChatClient.create(ollamaChatModel);
         */
    }
}
```

### Note

[illegible]

```
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);

    }

}
```

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

public record Answer(String answer) {
}
```

[illegible]

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

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.Question;

public interface QuestionService {

    String getAnswer(String question);

    Answer getAnswer(Question question);

}
```

### Note

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, typical of notebook paper. There are no margins, text, or other markings on the page.

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

import org.springframework.ai.chat.client.ChatClient;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Configuration;
import org.springframework.stereotype.Service;

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.Question;

@Service
@Configuration
public class QuestionServiceImpl implements QuestionService {

    private final ChatClient chatClient;

    public QuestionServiceImpl(@Qualifier("ollamaChatClient") ChatClient chatClient) {
        this.chatClient = chatClient;
    }

    @Override
    public String getAnswer(String question) {
        return this.chatClient.prompt()
            .user(question)
            .call()
            .content();
    }

    @Override
    public Answer getAnswer(Question question) {
        return new Answer(getAnswer(question.question()));
    }
}
```

### Note

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

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

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.Question;
import it.venis.ai.spring.demo.services.QuestionService;

@RestController

public class QuestionController {

    private final QuestionService service;

    public QuestionController(QuestionService service) {

        this.service = service;
    }

    @PostMapping("/client/ask")
    public Answer askQuestion(@RequestBody Question question) {

        return this.service.getAnswer(question);
    }
}
```

### Note

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



16 / 16

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.