

[illegible]

SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Settembre 2025

Note

[illegible]



SYSTEM E ROLE PROMPTING

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.

- ➡ *Focus*: impostare contesto, direttive, formato utilizzati dal LLM
- ⚠️ Lasciare alla parte *user* il solo compito di fornire l'istanza della *task* da risolvere
- ➡ Vantaggi
 - ➡ Aumento dell'efficacia con modelli contenuti
 - ➡ Possibilità di sfruttare un *system prompt* per molteplici *user prompts*
 - ➡ Maggior chiarezza ed ordine nella scrittura dei *prompt*

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. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface. There is no handwriting or other markings on the paper.

- ➔ Focus: assegnare un ruolo specifico al LLM
- ⚠ Specializzare la conversazione, utilizzando terminologia e registri tipici di specialisti del settore
- ➔ Vantaggi
 - ➔ Informazione generata in media più rilevante ed entropica

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. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

- 5 / 14

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.

Sei un esperto in materia di {artefatto}.

Rispondi in maniera coerente alla recensione fornita dallo '''user''', il cui sentiment è stato definito come {sentiment}.

In caso di recensione positiva, rispondi in maniera entusiasta e consiglia allo '''user''' altre tre scelte simili di {artefatto} legati a '''luoghi''' e '''persone''' riportati nella seguente lista YAML: {ner}. Se nella lista non ci sono '''luoghi''' o '''persone''', non dare suggerimenti di {artefatto}.

In caso di recensione negativa, rispondi in maniera garbata e professionale, scusandoti per la spiacevole esperienza.

In caso di recensione neutrale, informa lo '''user''' che rimarrai in attesa di ulteriori sue recensioni per poter consigliare nuove scelte di {artefatto}, chiedendo quali '''luoghi''' e '''persone''' lo hanno colpito.

[illegible]

```

Rispondi a questa recensione:
'''Titolo''': {titolo}
'''Sottotitolo''': {sottotitolo}
'''Recensione''': {corpo}

```

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.


```
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
```

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, providing a template for writing or drawing. There are no margins, text, or other markings on the paper.

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

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.ArtifactRequest;
import it.venis.ai.spring.demo.model.DefinitionRequest;
import it.venis.ai.spring.demo.model.DefinitionResponse;
import it.venis.ai.spring.demo.model.Question;

public interface GeminiFromClientService {

    String getAnswerFromClient(String question);

    Answer getAnswerFromClient(Question question);

    Answer getDefinitionFromClient(DefinitionRequest definitionRequest);

    Answer getCustomFormatDefinitionFromClient(DefinitionRequest definitionRequest);

    Answer getJSONUserFormatDefinitionFromClient(DefinitionRequest definitionRequest);

    DefinitionResponse getJSONOutputConverterFormatDefinitionFromClient(DefinitionRequest definitionRequest);

    Answer getSentimentForArtifact(ArtifactRequest artifactRequest);

    Answer getNERinYAMLForArtifact(ArtifactRequest artifactRequest);

    Answer getSuggestionForArtifact(ArtifactRequest artifactRequest);

}
```

Note

[illegible]

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

import java.util.Map;
import org.springframework.ai.chat.client.ChatClient;
import org.springframework.ai.template.st.StTemplateRenderer;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.core.io.Resource;
import org.springframework.stereotype.Service;
import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.DefinitionRequest;
import it.venis.ai.spring.demo.model.Question;

@Service
public class GeminiFromClientServiceImpl implements GeminiFromClientService {

    private final ChatClient chatClient;

    public GeminiFromClientServiceImpl(ChatClient.Builder chatClientBuilder) {
        this.chatClient = chatClientBuilder.build();
    }

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

    ...
}
```

Note

[illegible]

```

...
@Value("classpath:templates/set-advice-system-prompt.st")
private Resource adviceSystemPrompt;

@Value("classpath:templates/get-advice-user-prompt.st")
private Resource adviceUserPrompt;

@Override
public Answer getSuggestionForArtifact(ArtifactRequest artifactRequest) {
    String sentiment = getSentimentForArtifact(artifactRequest).answer();
    String ner = getNERinYAMLForArtifact(artifactRequest).answer();
    String chatResponse = this.chatClient.prompt()
        .options(ChatOptions.builder()
            .model("gemini-2.0-flash")
            .temperature(0.3)
            .topP(1.0)
            //.topK(30)
            .maxTokens(500)
            //.frequencyPenalty(0.1)
            //.presencePenalty(0.1)
            .build())
        .system(s -> s.text(this.adviceSystemPrompt)
            .params(Map.of("sentiment", sentiment, "ner", ner,
                "artefatto", artifactRequest.artifact().type()))
        .user(u -> u.text(this.adviceUserPrompt)
            .params(Map.of("titolo", artifactRequest.artifact().title(),
                "sottotitolo", artifactRequest.artifact().subtitle(),
                "corpo", artifactRequest.artifact().body())))
        .templateRenderer(StTemplateRenderer.builder().startDelimiterToken('{')
            .endDelimiterToken('}')
            .build())
        .call()
        .content();
    return new Answer(chatResponse);
}

```

Note

[illegible]

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

import org.springframework.web.bind.annotation.RestController;
...
import it.venis.ai.spring.demo.services.GeminiFromClientService;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;

@RestController
public class QuestionFromClientController {

    private final GeminiFromClientService geminiService;

    ...

    @PostMapping("/client/definition/json/user")
    public Answer getJSONUserFormatDefinition(@RequestBody DefinitionRequest definitionRequest) {

        return this.geminiService.getJSONUserFormatDefinitionFromClient(definitionRequest);

    }

    ...

    @PostMapping("/client/advice")
    public Answer getSuggestionForArtifact(@RequestBody ArtifactRequest artifactRequest) {

        return this.geminiService.getSuggestionForArtifact(artifactRequest);

    }

}
```

[illegible]

-  Simone Scannapieco

[illegible]

14 / 14

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.