



# 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

## FEW-SHOT PROMPTING

---

### ⇒ *Implicit prompting*

- 1 Test su REST API /ask per traduzione ITA/ENG, contrari, ...

### ⇒ *Few-shot prompting per NER extraction*

- 1 Creazione \*prompt template\* con esempi di estrazione NER da due articoli di giornale
- 2 Modifiche ad interfaccia ed implementazione del servizio Gemini
- 3 Modifica del controllore MVC per servizio Gemini

⚠ Il `path resources/templates` viene creato automaticamente da Spring AI in fase di creazione del progetto

### File `get-ner-yaml-prompt.st`

ESEMPIO:

```
'''Titolo''' La partigiana Rosani e il benefattore Forti, il rabbino: «Preghiamo per la pa  
'''Sottotitolo''' Le parole del rabbino Corinaldi al cimitero ebraico in occasione della c  
'''Corpo''' «Stiamo vivendo momenti importanti, che speriamo possano cambiare la situazione  
Oggi preghiamo per il popolo ebraico ma anche per tutto il mondo, a cui vogliamo bene.  
Cerchiamo di vivere in serenità, in pace».
```

```
Le parole sono del rabbino di Verona, Tomer Corinaldi, pronunciate al cimitero ebraico, in  
Oggi iniziano le 25 ore di digiuno totale e di preghiera per gli ebrei, in occasione dello  
«Confidiamo che potremo forse raggiungere la pace nella terra di Israele e per il popolo p  
Preghiamo per assicurare la sicurezza di tutti gli abitanti della terra.  
Gli ostaggi che stanno nel tunnel di Gaza da oltre due anni potrebbero finalmente tornare a  
Alla cerimonia è intervenuto anche l'assessore ai diritti umani, Jacopo Buffolo, ricordando  
«Rita come molte altre ragazze è caduta per la libertà e la pace», ha detto Buffolo, "ciò a  
«Le 25 ore di digiuno non sono lo scopo principale dello Yom Kippur, che è piuttosto un inv  
«Il nostro augurio è di una pace giusta e duratura».
```

```
'''YAML'''  
---
```

ner:

```
  persona:  
    - Tomer Corinaldi  
    - Achille Forti
```

### Interfaccia servizio Gemini

```
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);

}
```

### Implementazione servizio Gemini - I

```
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();
    }

    ...
}
```

## Implementazione servizio Gemini - II

```
...

@Value("classpath:templates/get-ner-yaml-prompt.st")
private Resource artifactNERYAMLPrompt;

@Override
public Answer getNERinYAMLForArtifact(ArtifactRequest artifactRequest) {

    String chatResponse = this.chatClient.prompt()
        .options(ChatOptions.builder()
            .model("gemini-2.0-flash")
            .temperature(0.1)
            .topP(1.0)
            //.topK(30)
            .maxTokens(250)
            //.frequencyPenalty(0.1)
            //.presencePenalty(0.1)
            .build())
        .user(u -> u.text(this.artifactNERYAMLPrompt)
            .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);
}
```



### Implementazione controllore REST

```
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/ner/yaml")
    public Answer getNERinYAML(@RequestBody ArtifactRequest artifactRequest) {

        return this.geminiService.getNERinYAMLForArtifact(artifactRequest);

    }

}
```

<https://github.com/simonescannapieco/spring-ai-base-dgroove-venis-code.git>  
*Branch:* 14-spring-ai-gemini-few-shot-prompting