



Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda



Note			



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Settembre 2025

Note		



Note	

PROGETTO SPRING AI PASSAGGI



- Sistema di traduzione automatica da italiano a multilingua
 - 1 Creazione modello TranslationRequest. java per richiesta traduzione lemma
 - Creazione prompt template per traduzione da italiano a multilingua
 - 3 Modifiche ad interfaccia ed implementazione del servizio Gemini
 - 4 Modifica del controllore MVC per servizio Gemini
 - Test delle funzionalità con Postman/Insomnia
- Sistema di sentiment analysis per artefatto
 - Creazione enumeratori ArtifactType.java e Sentiment.java
 - Creazione modello di artefatto Artifact. java
 - Creazione modello richiesta per artefatto ArtifactRequest.java
 - 4 Creazione prompt template per sentiment analysis di tipo 0-shot per artefatti
 - Modifiche ad interfaccia ed implementazione del servizio Gemini
 - 6 Modifica del controllore MVC per servizio Gemini
 - Test delle funzionalità con Postman/Insomnia

🐸 Simone Scannapieco	Spring AI - Corso base	<u>m</u> Venis S.p.A, Venezia, IT	3/

Note		



PROGETTO SPRING AI TRADUZIONE



Modello per richiesta traduzione

package it.venis.ai.spring.demo.model;
public record TranslationRequest(String lemma) {}

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

4/19

PROGETTO SPRING AI PROMPT PER TRADUZIONE MULTILINGUA



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

File get-lemma-translation-prompt.st

ITA:ciao->ENG:hello,ESP:hola,GER:hallo,FR:bonjour|ITA:casa->ENG:house,ESP:casa,GER:haus,FR:maison

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT €

5/19

Note		

Simone Scannapieco



6/19

m Venis S.p.A, Venezia, IT

```
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.AntifactRequest;
import it.venis.ai.spring.demo.model.DefinitionRequest;
import it.venis.ai.spring.demo.model.DefinitionRequest;
import it.venis.ai.spring.demo.model.Question;

public interface GeminiFromClientService {
    String getAnswerFromClient(String question);
    Answer getAnswerFromClient(Question question);
    Answer getOustomFormatDefinitionFromClient(DefinitionRequest);
    Answer getSONUserFormatDefinitionFromClient(DefinitionRequest definitionRequest);
    DefinitionResponse getJSONUsputConverterFormatDefinitionFromClient(DefinitionRequest definitionRequest definitionRequest);
    Answer getTranslationForLemma(TranslationRequest translationRequest);
}
```

Spring AI - Corso base

Note ______



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;
private final ChatClient chatClient;
   {\tt public\ GeminiFromClientServiceImpl(ChatClient.Builder\ chatClientBuilder)\ \{}
       this.chatClient = chatClientBuilder.build();
   public String getAnswerFromClient(String question) {
      return this.chatClient.prompt()
              .user(question)
              .call()
              .content();
```

🐸 Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

7/19





Implementazione servizio Gemini - II

```
@Value("classpath:templates/get-lemma-translation-prompt.st")
    private Resource lemmaTanslationPrompt;
    public Answer getTranslationForLemma(TranslationRequest translationRequest) {
         List<String> stopSequences = Stream.of(" ", "|", "\n").collect(Collectors.toList());
         String chatResponse = this.chatClient.prompt()
                   .options(ChatOptions.builder()
                   .model("gemini-2.0-flash")
                   .temperature(0.1)
                   //.topP(1.0)
//.topK(30)
                   .maxTokens(20)
                   //.frequencyPenalty(0.1)
//.presencePenalty(0.1)
.stopSequences(stopSequences)
                   .build())
                   .user(u -> u.text(this.lemmaTanslationPrompt)
                   .params(Map.of("lemma", translationRequest.lemma())))
.templateRenderer(StTemplateRenderer.builder().startDelimiterToken('{'})
                            .endDelimiterToken('}')
                            .build())
                   .call()
                   .content();
         return new Answer(chatResponse);
    }
}
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

8/19

PROGETTO SPRING AI MVC DEL SERVIZIO GEMINI



9/19

m Venis S.p.A, Venezia, IT

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/translate")
    public Answer getTranslationForLemma(@RequestBody TranslationRequest translationRequest) {
        return this.geminiService.getTranslationForLemma(translationRequest);
    }
}
```

Spring AI - Corso base

Ν	ote
---	-----

👺 Simone Scannapieco



Note	

PROGETTO SPRING AI TIPO DI ARTEFATTO



n

Enumeratore per tipo di artefatto

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

public enum ArtifactType {

    LIBRO("Libro"),
    FILM("Film"),
    SPETTACOLO("Spettacolo"),
    ARTICOLO("Articolo");

    private String artifactType;

    ArtifactType(String artifactType) {
        this.artifactType = artifactType;
    }

    public String getArtifactType() {
        return artifactType;
    }
}
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

10/19

PROGETTO SPRING AI SENTIMENT



Enumeratore per sentiment

```
package it.venis.ai.spring.demo.data;
public enum Sentiment {
    MOLTO_POSITIVO("molto positivo"),
    POSITIVO("positivo"),
NEUTRALE("neutrale"),
    NEGATIVO("negativo"),
MOLTO_NEGATIVO("molto negativo");
    private String sentiment;
    Sentiment(String sentiment) {
         this.sentiment = sentiment;
    public String getSentiment() {
        return sentiment;
}
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

11/19

PROGETTO SPRING AI ARTEFATTO



·

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

12/19

PROGETTO SPRING AI **ARTEFATTO**



Modello di richiesta per artefatto

```
package it.venis.ai.spring.demo.model;
import java.util.UUID;
public record ArtifactRequest(UUID id, Artifact artifact) {
    public ArtifactRequest(String title, Artifact artifact) {
    this(UUID.randomUUID(), artifact);
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

13/19

PROGETTO SPRING AI PROMPT PER SENTIMENT ANALYSIS





▲ II path resources/templates viene creato automaticamente da Spring AI in fase di creazione del progetto

File get-artifact-sentiment-prompt.st

Classifica recensioni di artefatti come |MOLTO_POSITIVO|POSITIVO|NEUTRALE| NEGATIVO | MOLTO_NEGATIVO | .

Restituisci la risposta usando la minima formattazione possibile, senza markdown e senza spazi finali.

Recensione {artefatto}: {recensione}

Sentimento:

Simone	Scanna	pieco
Cirrionic	Cournia	picoo

			_	
<i>≣</i> Sr	oring	AI -	Corso	bas

m Venis S.p.A, Venezia, IT

14/19

Note	

Simone Scannapieco



15/19

```
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.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 getUsstomFormatDefinitionFromClient(DefinitionRequest definitionRequest);

    Answer getUsstomFormatDefinitionFromClient(DefinitionRequest definitionRequest);

    DefinitionResponse getJSONOutputConverterFormatDefinitionFromClient(DefinitionRequest);

    Answer getTranslationForLemma(TranslationRequest translationRequest);

    Answer getSentimentForArtifact(ArtifactRequest artifactRequest);
}
```

m Venis S.p.A, Venezia, IT

Spring AI - Corso base

Note	



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;
private final ChatClient chatClient;
   {\tt public\ GeminiFromClientServiceImpl(ChatClient.Builder\ chatClientBuilder)\ \{}
       this.chatClient = chatClientBuilder.build();
   public String getAnswerFromClient(String question) {
      return this.chatClient.prompt()
              .user(question)
              .call()
              .content();
```

🐸 Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

16/19



Implementazione servizio Gemini - II

```
@Value("classpath:templates/get-artifact-sentiment-prompt.st")
   private Resource artifactSentimentPrompt;
   public Answer getSentimentForArtifact(ArtifactRequest artifactRequest) {
       List<String> stopSequences = Stream.of(" ", "\n").collect(Collectors.toList());
       Sentiment chatResponse = Sentiment.valueOf(this.chatClient.prompt()
               .options(ChatOptions.builder()
               .model("gemini-2.0-flash")
               .temperature(0.1)
               //.topP(1.0)
//.topK(30)
               .maxTokens(10)
               //.frequencyPenalty(0.1)
//.presencePenalty(0.1)
               .stopSequences(stopSequences)
               .build())
               .templateRenderer(StTemplateRenderer.builder().startDelimiterToken('{')
                       .endDelimiterToken('}')
                       .build())
               .call()
               .content()
       return new Answer(chatResponse.getSentiment());
}
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

17/19

PROGETTO SPRING AI MVC DEL SERVIZIO GEMINI



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/sentiment")
    public Answer getSentiment(@RequestBody ArtifactRequest artifactRequest) {
        return this.geminiService.getSentimentForArtifact(artifactRequest);
    }
}
```

Simone	Scanna	pieco
--------	--------	-------

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

18 / 19



https://github.com/simonescannapieco/spring-ai-base-dgroove-venis-code.git

Branch: 13-spring-ai-gemini-implicit-zero-shot-prompting

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

19 / 19