



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



- 1 Nuovo string template per richiesta definizione da dizionario in formato JSON specificato dall'utente
- Richiesta al portale Gemini di linee guida di eliminazione markdown da risposta in JSON
- 3 Modifica file GeminiFromClientService.java e GeminiFromClientServiceImpl.java
- 4 Modifica file QuestionFromClientController.java
- 5 Test delle funzionalità con Postman/Insomnia

Note		

PROGETTO SPRING AI MODELLO DI RICHIESTA



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

File get-json-user-format-definition-prompt.st

Dammi una definizione del termine <lemma>, come riporterebbe un dizionario. Restituisci la risposta in formato JSON, usando la minima formattazione possibile e senza markdown.

- Il JSON deve avere la seguente struttura:
 - il lemma deve essere inserito come valore della chiave '''lemma''';
 - la definizione in stile formale e conciso deve essere inserita come valore della chiave '''definizione''';
 - esempi di utilizzo del lemma come strutture JSON annidate sotto la chiave 'esempi'. Le strutture JSON annidate devono riportare come unico valore la '''frase''' di esempio.

臂 Simone	Scannapieco
----------	-------------

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

4/9

Note	

PROGETTO SPRING AI SERVIZIO GEMINI



Interfaccia servizio Gemini (da client)

```
package it.venis.ai.spring.demo.services;
import it.venis.ai.spring.demo.model.Answer;
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 getDefinitionFromClient(DefinitionRequest definitionRequest);

    Answer getCustomFormatDefinitionFromClient(DefinitionRequest definitionRequest);

    Answer getJSONUserFormatDefinitionFromClient(DefinitionRequest definitionRequest);
}
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

5/9

PROGETTO SPRING AI SERVIZIO GEMINI



Implementazione servizio Gemini (da client) - I

```
package it.venis.ai.spring.demo.services;
 import java.util.Map;
{\tt import org.springframework.ai.chat.client.ChatClient};
 {\tt import org.springframework.ai.template.st.StTemplateRenderer;}
 {\tt import org.springframework.beans.factory.annotation.Value;}
import org.springframework.core.io.Resource;
import org.springframework.stereotype.Service;
{\tt import\ it.venis.ai.spring.demo.model.Answer;}
{\tt import it.venis.ai.spring.demo.model.DefinitionRequest;}
import it.venis.ai.spring.demo.model.Question;
\verb"public class GeminiFromClientServiceImpl" implements GeminiFromClientService \{ in the content of the conten
              private final ChatClient chatClient;
              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

6/9

PROGETTO SPRING AI SERVIZIO GEMINI



Implementazione servizio Gemini (da client) - II

```
{\tt @Value("classpath:templates/get-json-user-format-definition-prompt.st")}
private Resource JSONUserFormatDefinitionPrompt;
ObjectMapper objectMapper;
public Answer getJSONUserFormatDefinitionFromClient(DefinitionRequest definitionRequest) {
    String chatResponse = this.chatClient.prompt()
             .user(u -> u.text(this.JSONUserFormatDefinitionPrompt)
                      .params(Map.of("lemma", definitionRequest.lemma())))
             .templateRenderer(StTemplateRenderer.builder().startDelimiterToken('<')</pre>
                      .endDelimiterToken('>')
                      .build())
             .call()
             .content();
    System.out.println(chatResponse);
    String responseString;
        JsonNode rootNode = objectMapper.readTree(chatResponse.replace("`","").replaceFirst("json","")); responseString = objectMapper.writeValueAsString(rootNode);
    } catch (JsonProcessingException e) {
        throw new RuntimeException(e);
    return new Answer(responseString);
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

7/9

PROGETTO SPRING AI MVC DEL SERVIZIO GEMINI



Implementazione controllore REST (da client)

```
package it.venis.ai.spring.demo.controllers;
 {\tt import org.springframework.web.bind.annotation.RestController;}
 import it.venis.ai.spring.demo.services.GeminiFromClientService;
 import org.springframework.web.bind.annotation.PostMapping;
{\tt import org.springframework.web.bind.annotation.RequestBody;}
public class QuestionFromClientController {
            private final GeminiFromClientService geminiService;
             @PostMapping("/client/definition")
            public Answer getDefinitionFromClient(@RequestBody DefinitionRequest definitionRequest) {
                        return this.geminiService.getDefinitionFromClient(definitionRequest);
            @PostMapping("/client/definition/custom")
            {\tt public \ Answer \ getCustomFormatDefinition(@RequestBody \ DefinitionRequest \ definitionRequest) \ \{ \ \ \ \} }
                         \textbf{return this}. \texttt{geminiService}. \texttt{getCustomFormatDefinitionFromClient} (\textbf{definitionRequest}); \\
            @PostMapping("/client/definition/json/user")
            {\tt public \ Answer \ getJSONUserFormatDefinition(@RequestBody \ DefinitionRequest \ definitionRequest) \ \{ to the least of the least 
                         \textbf{return this}. \texttt{geminiService}. \texttt{getJSONUserFormatDefinitionFromClient} (\textbf{definitionRequest}); \\
```

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

8/9



 $\label{lem:base-dgroove-venis-code.git} Branch:$

9-spring-ai-gemini-json-user-format-prompt-templates

Simone Scannapieco

Spring AI - Corso base

m Venis S.p.A, Venezia, IT

9/9

Note			