



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

CUSTOM PROMPT TEMPLATES

- 1 Nuovo *string template* per richiesta definizione da dizionario in formato utente
- 2 Modifica *file* `GeminiFromClientService.java` e `GeminiFromClientServiceImpl.java`
- 3 Modifica *file* `QuestionFromClientController.java`
- 4 Test delle funzionalità con Postman/Insomnia

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

File `get-custom-format-definition-prompt.st`

Dammi una definizione del termine `<lemma>`, come riporterebbe un dizionario.

La risposta deve essere restituita seguendo il formato:

`'''lemma scritto in maiuscolo e grassetto'''`.

Nella sua forma comune, `'''definizione in stile formale e conciso'''`.

Esempi di utilizzo: `'''esempi di utilizzo nella forma comune'''`.

Alcune accezioni: `'''accezioni con esempi'''`.

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);

}
```

Implementazione servizio Gemini (da client) - 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 (da client) - II

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

@Value("classpath:templates/get-definition-prompt.st")
private Resource definitionPrompt;

...

@Value("classpath:templates/get-custom-format-definition-prompt.st")
private Resource customFormatDefinitionPrompt;

@Override
public Answer getCustomFormatDefinitionFromClient(DefinitionRequest definitionRequest) {
    return new Answer(this.chatClient.prompt()
        .user(u -> u.text(this.customFormatDefinitionPrompt)
            .params(Map.of("lemma", definitionRequest.lemma()))
            .templateRenderer(StTemplateRenderer.builder().startDelimiterToken('<')
                .endDelimiterToken('>')
                .build())
            .call()
            .content());
}
}
```


Implementazione controllore REST (da *client*)

```
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;

    public QuestionFromClientController(GeminiFromClientService geminiService) {
        this.geminiService = geminiService;
    }

    @PostMapping("/client/ask")
    public Answer askQuestionFromClient(@RequestBody Question question) {
        return this.geminiService.getAnswerFromClient(question);
    }

    @PostMapping("/client/definition")
    public Answer getDefinitionFromClient(@RequestBody DefinitionRequest definitionRequest) {
        return this.geminiService.getDefinitionFromClient(definitionRequest);
    }

    @PostMapping("/client/definition/custom")
    public Answer getCustomFormatDefinition(@RequestBody DefinitionRequest definitionRequest) {
        return this.geminiService.getCustomFormatDefinitionFromClient(definitionRequest);
    }
}
```

<https://github.com/simonescannapieco/spring-ai-base-dgroove-venis-code.git>
Branch: 8-spring-ai-gemini-custom-format-prompt-templates