



# 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**

## **PROPRIETÀ EMERGENTI**

---

### ➔ *Embodiment*

- ➊ Modifica a *string template* per richiesta definizione da dizionario con formato di *output* automatizzato
- ➋ Modifica modello per richiesta a LLM (popolamento parametri di *default*)
- ➌ Modifica file `GeminiFromClientServiceImpl.java`
- ➍ Test delle funzionalità con Postman/Insomnia

### ➔ *Mathematics and Idiom understanding*

- ➊ Test delle funzionalità con Postman/Insomnia

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

**File** `get-json-output-converter-format-definition-prompt.st`

```
Dammi una definizione del termine {lemma}, come riporterebbe {agente}.  
{formato}
```

### File Modello di richiesta definizione

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

import java.util.Objects;
import java.util.UUID;

public record DefinitionRequest(UUID id, String lemma, String agent) {
    public DefinitionRequest(String lemma, String agent) {
        this(UUID.randomUUID(), lemma, agent);
    }

    @Override
    public String lemma() {
        return Objects.requireNonNullElse(this.lemma, "Ciao");
    }

    @Override
    public String agent() {
        return Objects.requireNonNullElse(this.agent, "Enciclopedia Treccani");
    }
}
```

### 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

```
...

@Value("classpath:templates/get-json-output-converter-format-definition-prompt.st")
private Resource JSONOutputConverterFormatDefinitionPrompt;

@Override
public DefinitionResponse getJSONOutputConverterFormatDefinitionFromClient(DefinitionRequest definitionRequest) {

    BeanOutputConverter<DefinitionResponse> converter = new BeanOutputConverter<>(DefinitionResponse.class);

    String format = converter.getFormat();

    System.out.println(format);

    String chatResponse = this.chatClient.prompt()
        .user(u -> u.text(this.JSONOutputConverterFormatDefinitionPrompt)
            .params(Map.of("lemma", definitionRequest.lemma(),
                "agente", definitionRequest.agent(),
                "formato", format)))
        .templateRenderer(StTemplateRenderer.builder().startDelimiterToken('{')
            .endDelimiterToken('}')
            .build())
        .call()
        .content();

    return converter.convert(Objects.requireNonNull(chatResponse));

}

}
```



<https://github.com/simonescannapieco/spring-ai-base-dgroove-venis-code.git>  
*Branch:* 11-spring-ai-gemini-emerging-capabilities