dgroove
Tecnologia emozionale.

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

🔧 ChatModel e ChatClient

Progetto Spring AI
Passaggi

1. Trasformazione *file* `GeminiService.java` in `GeminiFromModelService.java`
2. Trasformazione *file* `GeminiServiceImpl.java` in `GeminiFromModelServiceImpl.java`
3. Trasformazione *file* `QuestionController.java` in `QuestionFromModelController.java`
4. Creazione *file* `GeminiFromClientService.java` e `GeminiFromClientServiceImpl.java`
5. Creazione *file* `QuestionFromClientController.java`
6. Modifica *file* `GeminiServiceImplTest.java`
7. *Test* delle funzionalità con Postman/Insomnia

Progetto Spring AI
Servizio Gemini

Interfaccia servizio Gemini (da modello)

```java
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 GeminiFromModelService {

    String getAnswerFromModel(String question);

    Answer getAnswerFromModel(Question question);

    Answer getDefinitionFromModel(DefinitionRequest definitionRequest);

}
```

Implementazione servizio Gemini (da modello)

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

import java.util.Map;
import org.springframework.ai.chat.model.ChatModel;
import org.springframework.ai.chat.model.ChatResponse;
import org.springframework.ai.chat.prompt.Prompt;
import org.springframework.ai.chat.prompt.PromptTemplate;
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 GeminiFromModelServiceImpl implements GeminiFromModelService {

    private final ChatModel chatModel;

    public GeminiFromModelServiceImpl(ChatModel chatModel) {
        this.chatModel = chatModel;
    }

    @Override
    public String getAnswerFromModel(String question) {
        PromptTemplate promptTemplate = new PromptTemplate(question);
        Prompt prompt = promptTemplate.create();
        ChatResponse response = this.chatModel.call(prompt);
        return response.getResult().getOutput().getText();
    }

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

    @Value("classpath:templates/get-definition-prompt.st")
```

Implementazione controllore REST (da modello)

```java
package it.venis.ai.spring.demo.controllers;

import org.springframework.web.bind.annotation.RestController;
import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.DefinitionRequest;
import it.venis.ai.spring.demo.model.Question;
import it.venis.ai.spring.demo.services.GeminiFromModelService;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;

@RestController
public class QuestionFromModelController {

    private final GeminiFromModelService geminiService;

    public QuestionFromModelController(GeminiFromModelService geminiService) {
        this.geminiService = geminiService;
    }

    @PostMapping("/model/ask")
    public Answer askQuestion(@RequestBody Question question) {
        return this.geminiService.getAnswerFromModel(question);
    }

    @PostMapping("/model/definition")
    public Answer getDefinition(@RequestBody DefinitionRequest definitionRequest) {
        return this.geminiService.getDefinitionFromModel(definitionRequest);
    }

}
```

Progetto Spring AI
Servizio Gemini

Interfaccia servizio Gemini (da *client*)

```java
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);

}
```

Implementazione servizio Gemini (da *client*)

```
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();
    }

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

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

Implementazione controllore REST (da *client*)

```java
package it.venis.ai.spring.demo.controllers;

import org.springframework.web.bind.annotation.RestController;
import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.DefinitionRequest;
import it.venis.ai.spring.demo.model.Question;
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);
    }

}
```

Progetto Spring AI
Test servizio Gemini

Implementazione *test* Gemini

```java
package it.venis.ai.spring.demo.services;

import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
public class GeminiServiceImplTest {

    @Autowired
    GeminiFromModelServiceImpl geminiModelService;

    @Autowired
    GeminiFromClientServiceImpl geminiClientService;

    @Test
    void testGetAnswerFromModel() {
        String answer = geminiModelService.getAnswerFromModel("Raccontami una barzelletta");
        System.out.println(answer);
    }

    @Test
    void testGetAnswerFromClient() {
        String answer = geminiClientService.getAnswerFromClient("Spiegami cosa sei in una unica frase");
        System.out.println(answer);
    }
}
```

Codice

Branch di riferimento

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

Branch: `7-spring-ai-gemini-chatmodel-vs-chatclient`