

[illegible]

SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Novembre 2025

Note

[illegible]



CONCETTI AVANZATI

Note

[illegible]

- ## Interfaccia Chat Memory

 Simone Scannapieco

Note

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Interfaccia per ogni *chat memory* advisor

Customizzazione *id* conversazione

4 / 20

Note

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

- 1 Creazione modello per richiesta domanda con *username*
- 2 Modifica interfaccia e implementazione del servizio di risposta
- 3 Modifica controllore MVC
- 4 Test delle funzionalità con Postman/Insomnia

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice or general writing. There are no margins, text, or other markings on the paper.

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

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

public record QuestionRequest(UUID id, String username, Question body) {

    public QuestionRequest(String username, Question body) {

        this(UUID.randomUUID(), username, body);

    }

    @Override
    public String username() {

        return Objects.requireNonNullElse(this.username, "default");

    }

}
```

[illegible]

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

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.Question;
import it.venis.ai.spring.demo.model.QuestionRequest;

public interface QuestionService {

    Answer getGeminiAnswer(Question question);

    Answer getOllamaAnswer(Question question);

    Answer getOllamaDefaultAnswer(Question question);

    Answer getOllamaMemoryAwareAnswer(Question question);

    Answer getOllamaPerUserMemoryAwareAnswer(QuestionRequest request);

}
```

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.


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

...

@Service
@Configuration
public class QuestionServiceImpl implements QuestionService {

    private final ChatClient geminiChatClient;
    private final ChatClient ollamaChatClient;
    private final ChatClient ollamaMemoryChatClient;

    public QuestionServiceImpl(@Qualifier("geminiChatClient") ChatClient geminiChatClient,
                              @Qualifier("ollamaChatClient") ChatClient ollamaChatClient,
                              @Qualifier("ollamaMemoryChatClient") ChatClient ollamaMemoryChatClient) {

        this.geminiChatClient = geminiChatClient;
        this.ollamaChatClient = ollamaChatClient;
        this.ollamaMemoryChatClient = ollamaMemoryChatClient;
    }

    ...

    @Override
    public Answer getOllamaPerUserMemoryAwareAnswer(QuestionRequest request) {
        return new Answer(this.ollamaMemoryChatClient
            .prompt()
            .advisors(advisorSpec -> advisorSpec.param(ChatMemory.CONVERSATION_ID, request.username()))
            .user(request.body().question())
            .call()
            .content());
    }
}
```

Note

[illegible]

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

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.Question;
import it.venis.ai.spring.demo.model.QuestionRequest;
import it.venis.ai.spring.demo.services.QuestionService;

@RestController
public class QuestionController {

    ...

    @PostMapping("/ollama/ask/memory")
    public Answer getOllamaMemoryAwareAnswer(@RequestBody Question question) {

        return this.service.getOllamaMemoryAwareAnswer(question);
    }

    @PostMapping("/ollama/ask/memory/user")
    public Answer getOllamaPerUserMemoryAwareAnswer(@RequestBody QuestionRequest request) {

        return this.service.getOllamaPerUserMemoryAwareAnswer(request);
    }
}
```

[illegible]

10 / 20

[illegible]

- ## Dipendenza di progetto

 Simone Scannapieco

Note

[illegible]

```
@Autowired
JdbcChatMemoryRepository chatMemoryRepository;

ChatMemory chatMemory = MessageWindowChatMemory.builder().chatMemoryRepository(chatMemoryRepository)
    .maxMessages(10)
    .build();
```

- ```
spring:
 datasource:
 url: jdbc:postgresql:...
```

- ```
CREATE TABLE IF NOT EXISTS SPRING_AI_CHAT_MEMORY (
    conversation_id VARCHAR(36) NOT NULL,
    content TEXT NOT NULL,
    type VARCHAR(10) NOT NULL CHECK (type IN ('USER', 'ASSISTANT', 'SYSTEM', 'TOOL')),
    "timestamp" TIMESTAMP NOT NULL
);

CREATE INDEX IF NOT EXISTS SPRING_AI_CHAT_MEMORY_CONVERSATION_ID_TIMESTAMP_IDX
ON SPRING_AI_CHAT_MEMORY(conversation_id, "timestamp");
```

Note

[illegible]

File ./resources/schema/schema-h2.sql

```
CREATE TABLE SPRING_AI_CHAT_MEMORY (
    conversation_id VARCHAR(36) NOT NULL,
    content LONGVARCHAR NOT NULL,
    type VARCHAR(10) NOT NULL,
    "timestamp" TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL
);

CREATE INDEX SPRING_AI_CHAT_MEMORY_CONVERSATION_ID_TIMESTAMP_IDX
ON SPRING_AI_CHAT_MEMORY(conversation_id, timestamp DESC);

ALTER TABLE SPRING_AI_CHAT_MEMORY ADD CONSTRAINT TYPE_CHECK CHECK (type IN ('USER', 'ASSISTANT', 'SYSTEM', 'TOOL'));
```

Note

[illegible]

2 Impostare i corretti parametri dell'applicativo

File application.yml

```
spring:
  ai:
    chat:
      ...
      memory:
        repository:
          jdbc:
            initialize-schema: always # always, never, embedded (default)
            schema: classpath:/schema/schema-h2.sql
    ...
datasource:
  url: jdbc:h2:file:./db/chatmemory
  driver-class-name: org.h2.Driver
  username: user
  password: pwd
h2:
  console:
    enabled: true
...

```

File pom.xml

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <optional>true</optional>
</dependency>

<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

Note

[illegible]

- 1 Modifica al `pom.xml` per dipendenze Spring AI JDBC, H2 e *devtools*
- 2 Modifica ad `application.yml`
- 3 Creazione schema H2
- 4 Modifica ai *bean* `ChatClient` per Ollama
- 5 Test delle funzionalità con Postman/Insomnia

[illegible]


```

<..
<dependency>
  <groupId>org.springframework.ai</groupId>
  <artifactId>spring-ai-starter-model-chat-memory-repository-jdbc</artifactId>
</dependency>
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <optional>true</optional>
</dependency>
<..

```

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```
spring:
  ai:
    chat:
      ...
      memory:
        repository:
          jdbc:
            initialize-schema: always # always, never, embedded (default)
            schema: classpath:/schema/schema-h2.sql
      ...
datasource:
  url: jdbc:h2:file:./db/chatmemory
  driver-class-name: org.h2.Driver
  username: user
  password: pwd
h2:
  console:
    enabled: true
...

```

[illegible]

```
CREATE TABLE SPRING_AI_CHAT_MEMORY (
    conversation_id VARCHAR(36) NOT NULL,
    content LONGVARCHAR NOT NULL,
    type VARCHAR(10) NOT NULL,
    "timestamp" TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL
);

CREATE INDEX SPRING_AI_CHAT_MEMORY_CONVERSATION_ID_TIMESTAMP_IDX
ON SPRING_AI_CHAT_MEMORY(conversation_id, timestamp DESC);

ALTER TABLE SPRING_AI_CHAT_MEMORY ADD CONSTRAINT TYPE_CHECK CHECK (type IN ('USER', 'ASSISTANT', 'SYSTEM', 'TOOL'));
```

Note

[illegible]

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

...

@Configuration
public class MemoryChatClientConfig {

    ...

    /**
     * Bean to store messages into relational DBs. This is done
     * under the hood when the corresponding .pom dependency is added.
     * The system infers the type of DB through properties in application
     * .properties/.yml file.
     */
    @Autowired
    JdbcChatMemoryRepository jdbcChatMemoryRepository;

    /**
     * Custom bean for chat memory, based on the (auto-)configured chat memory repository.
     * Done under-the-hood via Spring auto-configuration with maxMessages = 20.
     */
    @Bean
    public ChatMemory chatMemory(@Qualifier("jdbcChatMemoryRepository") ChatMemoryRepository chatMemoryRepository) {
        return MessageWindowChatMemory.builder()
            .chatMemoryRepository(chatMemoryRepository)
            .maxMessages(20)
            .build();
    }

    ...

}
```

Note

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There are no vertical margin lines, and the paper is otherwise completely blank.



English

PreferencesToolsHelp

Login

Saved Settings:

Generic H2 (Embedded)

Setting Name:

Generic H2 (Embedded)

Save

Remove

Driver Class:

org.h2.Driver

JDBC URL:

jdbc:h2:~test

User Name:

sa

Password:

Connect

Test Connection

[illegible]

Branch: 6-spring-ai-gemini-ollama-jdbc-chat-memory

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.