



Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda



Note			



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

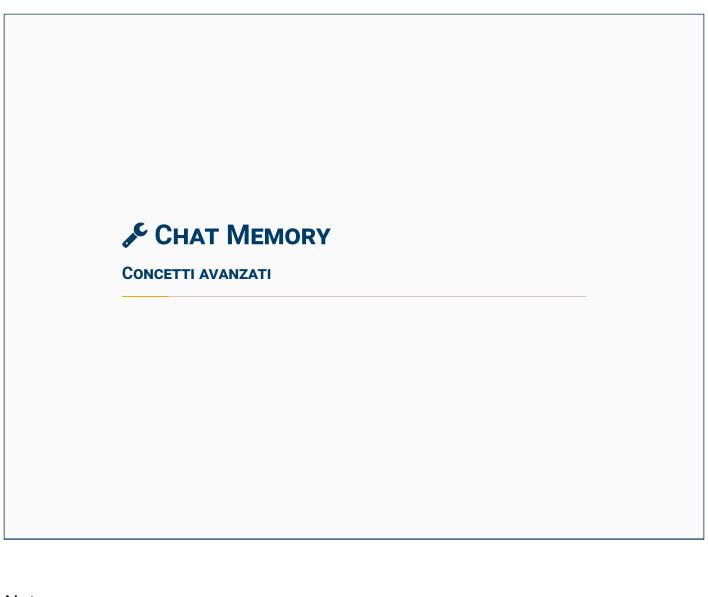
Novembre 2025

🐸 Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

Note			
_			



Note		

CONCETTI AVANZATI PER-USER CHAT MEMORY



- Ocome tenere traccia di diverse conversazioni?
 - ChatMemory definita secondo una logica multi-conversazione
 - ▲ CONVERSATION_ID chiave che il sistema ricerca nel contesto per CRUD storico messaggistica
 - DEFAULT_CONVERSATION_ID valore di default fornito alla chiave se non popolata programmaticamente

Interfaccia Chat Memory

```
public interface ChatMemory {
   String DEFAULT_CONVERSATION_ID = "default";
   /**
   * The key to retrieve the chat memory conversation id from the context.
   */
   String CONVERSATION_ID = "chat_memory_conversation_id";
   ...
```

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

3/20

CONCETTI AVANZATI PER-USER CHAT MEMORY



- Come sovrascrivere il valore di default?
 - Agire a livello di chat memory advisor
 - Utilizzo delle advisor specifications di ChatClient (ChatClient\$AdvisorSpec)

Interfaccia per ogni chat memory advisor

Customizzazione id conversazione

```
String conversation_id = "custom_conversation_id";
ChatClient
    .prompt()
    .advisors(advisorSpec -> advisorSpec.param(ChatMemory.CONVERSATION_ID, conversation_id))
    .build();
```

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

4/20

PROGETTO SPRING AI APPLICAZIONE E PASSAGGI



- Onfigurazione per-user chat memory per ChatClient Ollama
 - 1 Creazione modello per richiesta domanda con username
 - Modifica interfaccia e implementazione del servizio di risposta
 - Modifica controllore MVC
 - 4 Test delle funzionalità con Postman/Insomnia

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

Note			



Modello richiesta con username 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"); } }

👺 Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

6/20

👺 Simone Scannapieco



m Venis S.p.A, Venezia, IT

7/20

Interfaccia servizio 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.Question; 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); }

Spring AI - Corso avanzato

Note ______



Implementazione servizio

```
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\ Question Service Impl(@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
                 .advisors(advisorSpec -> advisorSpec.param(ChatMemory.CONVERSATION_ID, request.username()))
                 .user(request.body().question())
                 .content());
```

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

8/20



9/20

m Venis S.p.A, Venezia, IT

Implementazione controllore REST

```
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;

ORestController
public class QuestionController {

...

OPostMapping("/ollama/ask/memory")
public Answer getOllamaMemoryAwareAnswer(QRequestBody Question question) {
    return this.service.getOllamaMemoryAwareAnswer(question);
}

OPostMapping("/ollama/ask/memory/user")
public Answer getOllamaPerUserMemoryAwareAnswer(GRequestBody QuestionRequest request) {
    return this.service.getOllamaPerUserMemoryAwareAnswer(request);
}
```

Spring AI - Corso avanzato

N	ot	e

🐸 Simone Scannapieco



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

Branch: 5-spring-ai-gemini-ollama-per-user-chat-memory

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

Ν	ot	e
---	----	---

CONCETTI AVANZATI PERSISTENT CHAT MEMORY



- ▲ Storico della conversazione valida per singola sessione
 - Opzione valida per demo/applicazioni a bassa criticità
 - Possibile persistere le informazioni di storico su DB
 - PostgreSQL
 - MySQL/MariaDB
 - SQL Server
 - HSQLDB

Dipendenza di progetto

<dependency>

<groupId>org.springframework.ai</groupId>

<artifactId>spring-ai-starter-model-chat-memory-repository-jdbc</artifactId>

</dependency>

O.	Simone	Scanna	apieco
	SILLIOLIE	Scaring	apiect

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

Note			





Definizione bean

```
@Autowired
JdbcChatMemoryRepository chatMemoryRepository;
{\tt ChatMemory \ chatMemory = MessageWindowChatMemory.builder().chatMemoryRepository(chatMemoryRepository)}
    .maxMessages(10)
```

Il sistema inferisce la tipologia di DB dalle proprietà dell'applicativo

File application.yml

```
datasource:
    url: jdbc:postgresql:...
```

Il sistema applica lo schema relativo al DB inferito

File schema-postgresql.sql

```
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");
```

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

12/20

CONCETTI AVANZATI DEFINIZIONE NUOVO CONNETTORE

Simone Scannapieco



13/20

m Venis S.p.A, Venezia, IT

1 Definire un nuovo schema per il nuovo connettore

```
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	

Spring AI - Corso avanzato

CONCETTI AVANZATI DEFINIZIONE NUOVO CONNETTORE



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
       url: jdbc:h2:file:./db/chatmemory
       driver-class-name: org.h2.Driver
       username: user
       password: pwd
   h2:
       console:
           enabled: true
```

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

13 / 20

PROGETTO SPRING AI APPLICAZIONE E PASSAGGI



14/20

- Onfigurazione JDBC H2 chat memory per ChatClient Ollama
 - 1 Modifica al pom.xml per dipendenze Spring Al 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

Note		

PROGETTO SPRING AI JDBC CHAT MEMORY

🐸 Simone Scannapieco



m Venis S.p.A, Venezia, IT

15/20

Spring AI - Corso avanzato

Note	

PROGETTO SPRING AI JDBC CHAT MEMORY





Configurazione applicativo spring: ai: chat: memory:

```
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
```

password: pwd username: user console:

repository:

enabled: true

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

16/20

PROGETTO SPRING AI JDBC CHAT MEMORY

👺 Simone Scannapieco



m Venis S.p.A, Venezia, IT

17/20

```
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		

Spring AI - Corso avanzato

PROGETTO SPRING AI JDBC CHAT MEMORY



Configurazione Gemini + Ollama 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 correponding .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. */

 ${\tt public\ ChatMemory} \ (\texttt{Qualifier}(\texttt{"jdbcChatMemoryRepository"})\ ChatMemoryRepository\ chatMemoryRepository)\ \{\texttt{ChatMemoryRepository}, \texttt{ChatMemoryRepository}, \texttt{$

Simone Scannapieco

}

Spring AI - Corso avanzato

return MessageWindowChatMemory.builder()

.maxMessages(20)
.build();

.chatMemoryRepository(chatMemoryRepository)

m Venis S.p.A, Venezia, IT

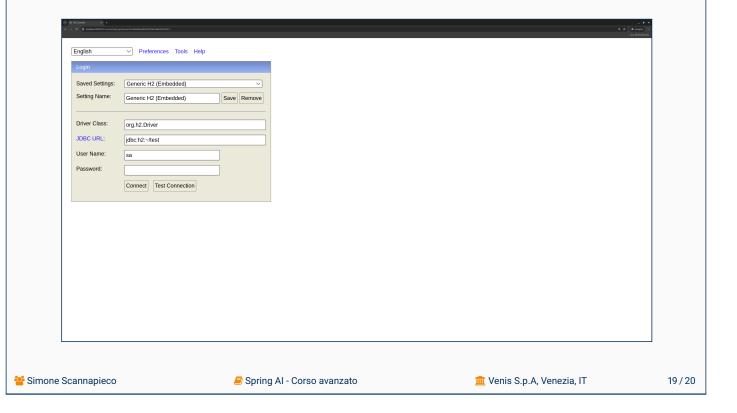
18/20

AMBIENTE DI SVILUPPO **POPOLAMENTO VECTOR STORE**





△ Verificare la dashboard H2 (http://localhost:8080/h2-console)



Note		



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

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

Simone Scannapieco

Spring AI - Corso avanzato

m Venis S.p.A, Venezia, IT

ot	e
	ot