



ICT Training Center

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



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.

SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Novembre 2025

Note

[illegible]

-  Simone Scannapieco

[illegible]

 Simone Scannapieco
 Spring AI - Corso avanzato
 Venis S.p.A, Venezia, IT
 4 / 7

This image shows a single 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.

```
@Component
public class TimeTools {

    @Tool(name="getCurrentLocalTime",
        description="Ottieni l'ora corrente nel fuso orario dell'utente")
    String getCurrentLocalTime() {
        ...
    }

    @Tool(name="getCurrentTime",
        description="Ottieni l'ora corrente nel fuso orario specificato.")
    public String getCurrentTime(@ToolParam(description = "Valore che rappresenta il fuso orario") String timeZone) {
        ...
    }
}
```

- Simone Scannapieco

Note

[illegible]

```
@Component
public class TimeTools {

    @Tool(name="getCurrentLocalTime",
        description="Ottieni l'ora corrente nel fuso orario dell'utente")
    String getCurrentLocalTime() {
        ...
    }

    @Tool(name="getCurrentTime",
        description="Ottieni l'ora corrente nel fuso orario specificato.")
    public String getCurrentTime(@ToolParam(description = "Valore che rappresenta il fuso orario") String timeZone) {
        ...
    }
}
```

Processo di *method as tool*

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.

```
@Bean
public ChatClient ollamaChatClient(OllamaChatModel ollamaChatModel, TimeTools timeTools) {

    ChatClient.Builder chatClientBuilder = ChatClient.builder(ollamaChatModel);

    return chatClientBuilder
        .defaultTools(timeTools)
        ...
        .build();
}
```



Note

[illegible]

-
- The diagram illustrates the Spring AI Tool Dispatching Architecture, showing the flow of data and control between various components:
- Chat Request** (Left): Contains a **Tool Definition** (Green box). It lists attributes: `name`, `description`, and `input schema`.
 - Spring AI** (Center): A large container housing the **Dispatch Tool Call Requests** component (White box with yellow border).
 - Tool** (Top): A green box representing the tool being dispatched.
 - AI Model** (Bottom): A yellow box representing the AI model that processes requests.
 - Chat Response** (Right): The final output of the process.
- The flow is numbered 1 through 6:
- 1**: A solid arrow from **Tool Definition** to **Dispatch Tool Call Requests**.
 - 2**: A solid arrow from **Dispatch Tool Call Requests** to **AI Model**.
 - 3**: A solid arrow from **AI Model** to **Tool**.
 - 4**: A solid arrow from **Tool** to **Dispatch Tool Call Requests**.
 - 5**: A solid arrow from **Dispatch Tool Call Requests** to **AI Model**.
 - 6**: A solid arrow from **Dispatch Tool Call Requests** to **Chat Response**.
- Dashed arrows indicate feedback loops from the **AI Model** back to the **Tool** (via step 3) and back to the **Dispatch Tool Call Requests** (via step 5).

 Simone Scannapieco

Spring AI - Corso avanzato

 Venis S.p.A, Venezia, IT

7/7

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.

- 2

[illegible]

- 3

[illegible]

-
- The diagram illustrates the Spring AI Tool Call Flow, showing the interaction between a Chat Request, Spring AI, a Tool, and an AI Model.
- Components:**
- Chat Request:** Contains a **Tool Definition** (green box) with attributes: `name`, `description`, and `input schema`.
 - Spring AI:** Contains a **Dispatch Tool Call Requests** component (white box with yellow border).
 - Tool:** A green box representing the external tool.
 - AI Model:** A yellow box representing the AI model.
 - Chat Response:** The final output of the process.
- Flow Steps:**
- 1:** The **Tool Definition** is passed from the **Chat Request** to the **Dispatch Tool Call Requests** component within **Spring AI**.
 - 2:** The **Dispatch Tool Call Requests** component sends a request to the **AI Model**.
 - 3:** The **AI Model** returns a response to the **Dispatch Tool Call Requests** component.
 - 4:** The **Dispatch Tool Call Requests** component sends a request to the **Tool**.
 - 5:** The **Tool** returns a response to the **Dispatch Tool Call Requests** component.
 - 6:** The **Dispatch Tool Call Requests** component sends the final **Chat Response**.
- Feedback Loops:**
- A dashed arrow from the **AI Model** back to the **Dispatch Tool Call Requests** component (Step 2 to 3).
 - A dashed arrow from the **Tool** back to the **Dispatch Tool Call Requests** component (Step 4 to 5).

 Simone Scannapieco

Spring AI - Corso avanzato

 Venis S.p.A, Venezia, IT

717

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.

-
- The diagram illustrates the Spring AI Tool Call Flow, showing the interaction between a Chat Request, Spring AI, a Tool, and an AI Model.
- Components:**
- Chat Request:** Contains a **Tool Definition** (green box) with attributes: `name`, `description`, and `input schema`.
 - Spring AI:** Contains a **Dispatch Tool Call Requests** component (white box with yellow border).
 - Tool:** A green box representing the external tool.
 - AI Model:** A yellow box representing the AI model.
 - Chat Response:** The final output of the process.
- Flow Steps:**
- 1:** The **Tool Definition** is passed from the **Chat Request** to the **Dispatch Tool Call Requests** component within **Spring AI**.
 - 2:** The **Dispatch Tool Call Requests** component sends a request to the **AI Model**.
 - 3:** The **AI Model** returns a response to the **Dispatch Tool Call Requests** component.
 - 4:** The **Dispatch Tool Call Requests** component sends a request to the **Tool**.
 - 5:** The **Tool** returns a response to the **Dispatch Tool Call Requests** component.
 - 6:** The **Dispatch Tool Call Requests** component sends the final **Chat Response**.
- Feedback Loops:**
- A dashed arrow from the **AI Model** back to the **Dispatch Tool Call Requests** component (Step 2 to 3).
 - A dashed arrow from the **Tool** back to the **Dispatch Tool Call Requests** component (Step 4 to 5).

 Simone Scannapieco

Spring AI - Corso avanzato

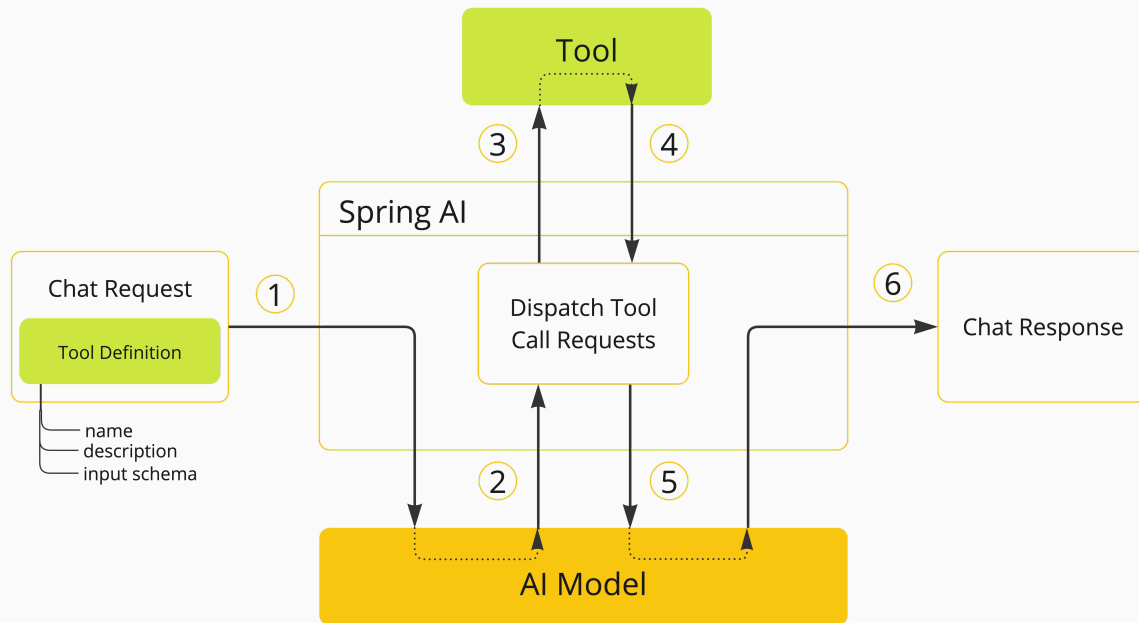
 Venis S.p.A, Venezia, IT

717

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.

- 6** Il LLM restituisce la risposta finale al `ChatClient/ChatModel` sfruttando generazione di testo e contesto



©Spring AI

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.