Build AI Applications with Spring AI

FU CHENG

Build AI Applications with Spring AI

RAG, MCP and Agents with Spring AI 1.1

Fu Cheng

This book is available at https://leanpub.com/spring-ai

This version was published on 2025-12-31

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2025 Fu Cheng

Also by single authorAlso By Fu Cheng

Exploring Java 25
Text-to-SQL, Spring AI Implementation with RAG
Understanding Java Virtual Threads
From Java 21 to Java 25
From Java 17 to Java 21
Build Native Java Apps with GraalVM
From Java 11 to Java 17
ES6 Generators
A Practical Guide for Java 8 Lambdas and Streams
Lodash 4 Cookbook
JUnit 5 Cookbook

Contents

Spring AI Basics 1

Getting Started 2
  Prerequisites 2
  Spring Boot Application 4
  Use Model Service 7
  Consolidate Local and Production Environment 7

Chat Completion 13
  Model 13
  ChatModel 13
  Create Prompts 13
  Chat Response 14
  ChatClient 14
  Use Custom Chat Options 14
  Prompt Template 15
  Advisor 15

Streaming Chat Completion 18
  StreamingChatModel 18
  Streaming Web Application 18
  Streaming JSON Events 18

Structured Output Converter 19
  StructuredOutputConverter 19
  ListOutputConverter 19
  MapOutputConverter 19
  BeanOutputConverter 19
  Use ChatClient 19

Multimodal Input 20
  Media 20

Image Understanding 20

Retrieval-Augmented Generation **21**

RAG Introduction **22**
  Reduce Hallucinations 22
  Naive RAG 24

Embedding Model **27**
  `EmbeddingModel` 27
  `EmbeddingOptions` 27
  `BatchingStrategy` 27
  Use `EmbeddingModel` 27

Documents **28**
  Create Documents 28
  Document Reader 28
  Document Transformer 28
  Document Writer 28

Vector Store **29**
  VectorStore 29
  Create `VectorStore` 29
  Add Documents 29
  Delete Documents 29
  Similarity Search 29
  VectorStore REST API 29
  SimpleVectorStore 30
  Pgvector 30
  Vector Store Cloud Services 30

RAG **32**
  Simple RAG 32

Modular RAG **33**
  Query 33
  Pre-Retrieval 33
  Retrieval 34
  Post-Retrieval 34
  Generation 35

RetrievalAugmentationAdvisor 35

RAG Examples **36**
Text-to-SQL 36
PDF Q&A 37

MCP 39

MCP Introduction **40**

Quick Start **41**
Java Development Basics 41
stdio Server 41
HTTP SSE Server 41
MCP Client 41
Spring Integration 41
Spring AI Integration 42

MCP Server **43**
Shared Models 43
ServerExchange 43
Prompt Templates 43
Resources 43
Tools 44
Completions 44
Logging 44
Pagination 44

MCP Client **45**
Roots 45
Sampling 45

MCP Examples **46**
File System MCP Server 46

Agent 47

Agent Introduction **48**
Cooking Suggestion Agent 48

Agent Components **49**
  Profile 49
  Persona 49
  Tools 49
  Knowledge and memory 49
  Reasoning and evaluation 49
  Planning and feedback 49

Agentic Patterns **51**
  Task Execution 51
  Evaluator-Optimizer 52
  Parallelization Workflow 53
  Routing Workflow 53
  Chain Workflow 54
  Agent as Tool 54
  Tool as Agent 55
  Orchestrator-Workers Workflow 55

Agent Development **56**
  Persona 56
  Knowledge 56
  Tools 56
  Reasoning 56
  Task 56

Source Code and Materials **57**

Spring AI Basics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Getting Started

Let’s start the journey with Spring AI from a simple application.

Prerequisites

Before writing Spring AI applications, we need to prepare the local development environment. Obviously, we need to have Java installed and configured. We also need to have a large language model (LLM) ready for testing.

Java

Spring AI requires a minimal Java version of 17. It’s recommended to use Java 21 or Java 25 LTS, so we can leverage the power of virtual threads.

Source code of this book is tested using Java 21 with virtual threads enabled.

Spring AI

This book uses Spring AI `1.1.0`. Example applications in this book use Maven to manage dependencies.

To simplify dependency management of related modules, the `spring-ai-bom` dependency can be imported to set versions of Spring AI dependencies.

Figure 1. Spring AI dependency management using BOM

```xml
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.ai</groupId>
      <artifactId>spring-ai-bom</artifactId>
      <version>${spring-ai.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Language Model

A language model is required for development, testing and production deployments. This language model can run locally or on the cloud, as long as it provides an API endpoint to access its service.

- To run a model locally, there are many options available, including Ollama, vLLM, and LM Studio.
- To use a cloud-based model service, you need to open an account and pay the service by tokens.

Here let's start from using Ollama.

Ollama is a tool to run large language models locally. You can simply download Ollama and install it on your local machine. After installation, you can open a terminal window and use Ollama CLI command `ollama` to work with it.

There are many models available for use with Ollama, see Ollama's models page for a full list.

We can use `ollama pull` to pull a model. Here we are using Qwen3.

Figure 2. Ollama pull a model

```
ollama pull qwen3:0.6b
```

The size of `qwen3:0.6b` is only 523MB. It's good for local development and testing.

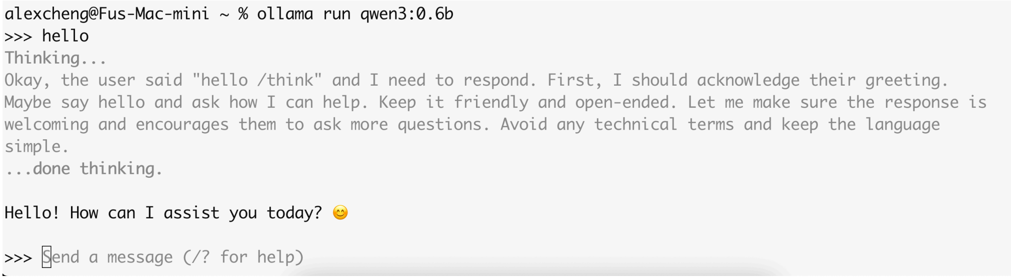
After the model is pulled, it can be run using `ollama run`.

Figure 3. Ollama run a model

```
ollama run qwen3:0.6b
```

`ollama run` command pulls non-existing models automatically.

`ollama run` starts a command-line session with the LLM. You can simply type any text to receive completions from LLM.



```
alexcheng@Fus-Mac-mini ~ % ollama run qwen3:0.6b
>>> hello
Thinking...
Okay, the user said "hello /think" and I need to respond. First, I should acknowledge their greeting.
Maybe say hello and ask how I can help. Keep it friendly and open-ended. Let me make sure the response is
welcoming and encourages them to ask more questions. Avoid any technical terms and keep the language
simple.
...done thinking.

Hello! How can I assist you today? 😊

>>> Send a message (/? for help)
```

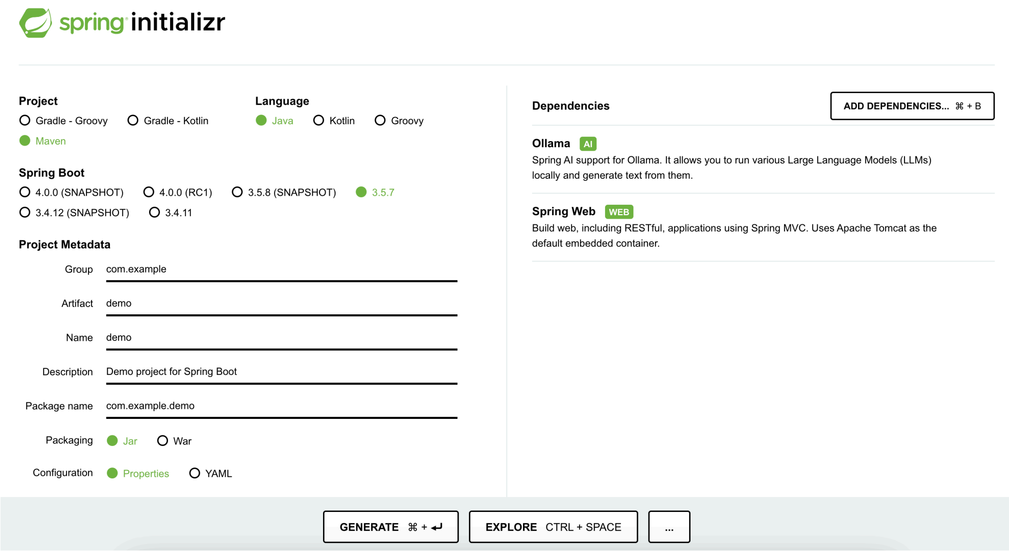
Figure 4. ollama run

By default, Ollama provides its API endpoint at port `11434`.

Spring Boot Application

The easiest way to create a new Spring AI application is using Spring Initializr. When adding the project's dependencies, `Ollama` should be selected. This enables Spring AI to interact with Ollama. `Spring Web` is also added to create a simple REST API.

Below is the screenshot of Spring Initializr.



Project
○ Gradle - Groovy ○ Gradle - Kotlin
● Maven

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 4.0.0 (SNAPSHOT) ○ 4.0.0 (RC1) ○ 3.5.8 (SNAPSHOT) ● 3.5.7
○ 3.4.12 (SNAPSHOT) ○ 3.4.11

Project Metadata

Group: com.example
Artifact: demo
Name: demo
Description: Demo project for Spring Boot
Package name: com.example.demo
Packaging: ● Jar ○ War
Configuration: ● Properties ○ YAML

Dependencies

Ollama AI
Spring AI support for Ollama. It allows you to run various Large Language Models (LLMs) locally and generate text from them.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Figure 5. Spring Initializr UI

Now we can download the created application and open it using IntelliJ IDEA.

Adding the `Ollama` dependency actually includes the `spring-ai-starter-model-ollama` to the Maven project. This Spring Boot starter will create necessary beans to work with Spring AI.

Figure 6. Spring Boot Ollama starter

```xml
<dependency>
  <groupId>org.springframework.ai</groupId>
  <artifactId>spring-ai-starter-model-ollama</artifactId>
</dependency>
```

Here we need to add an `application.yaml` file to configure the Spring Boot application. This is because `qwen3` model should be used. By default, Ollama uses Mistral model. The property to configure the Ollama model is `spring.ai.ollama.chat.options.model`.

Figure 7. Spring Boot configuration for Spring AI using Ollama

```yaml
spring:
  ai:
    ollama:
      chat:
        options:
          model: "qwen3:0.6b"
```

Now we add a REST endpoint to chat with an LLM. The `ChatClient.Builder` instance is injected into the REST controller to create `ChatClient` instances. This instance is provided by Ollama Spring Boot starter. A `ChatClient` is created from this `ChatClient.Builder` using the `build` method. `chatClient.prompt().user(message).call().content()` sends a request to Ollama API endpoint and receives the output.

Figure 8. REST Controller

```java
@RestController
public class ChatController {

  private final ChatClient chatClient;

  public ChatController(ChatClient.Builder builder,
      LoggingAdvisor loggingAdvisor) {
    this.chatClient = builder.defaultAdvisors(loggingAdvisor).build();
  }

  @GetMapping("/chat")
  public String chat(@RequestParam(value = "message") String message) {
    return chatClient.prompt().user(message).call().content();
  }
}
```

Now we can start the Spring Boot application. Once the application is started, we can use any REST client tool to interact with the REST API.

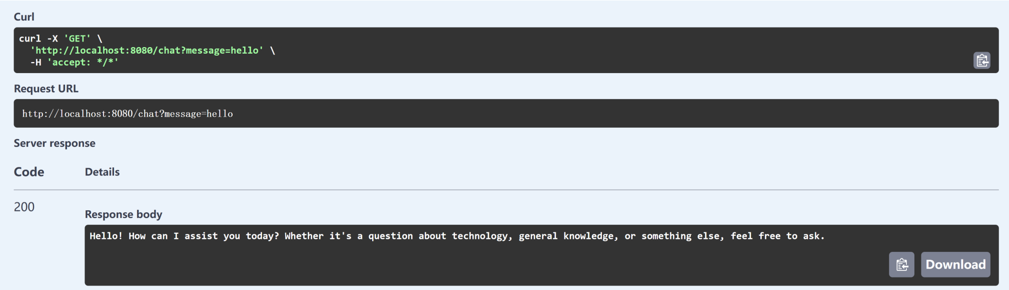
Here we use SpringDoc to expose OpenAPI endpoint and Swagger UI to test the API.

Figure 9. SpringDoc dependency

```xml
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.8.9</version>
</dependency>
```

We can open a browser window and navigate to `http://localhost:8080/swagger-ui/`, then use Swagger UI to try the API.

Below is the result of testing the API using Swagger UI.



```
curl -X 'GET' \
  'http://localhost:8080/chat?message=hello' \
  -H 'accept: */*'
```

```
http://localhost:8080/chat?message=hello
```

200

```
Hello! How can I assist you today? Whether it's a question about technology, general knowledge, or something else, feel free to ask.
```

Figure 10. Use Swagger UI

Use Model Service

While Ollama is great for local development and testing, we usually use cloud-based model services for production. All major cloud platforms provide AI models as services, including Google, Amazon, and Microsoft. Spring AI supports major AI model services. Here OpenAI is used as an example.

For Spring Boot, the easiest way is adding the Spring Boot starter dependency. For OpenAI support, the dependency is `spring-ai-starter-model-openai`.

Figure 11. OpenAI Spring Boot starter dependency

```xml
<dependency>
  <groupId>org.springframework.ai</groupId>
  <artifactId>spring-ai-starter-model-openai</artifactId>
</dependency>
```

An OpenAI API key is required to use OpenAI services. In the configuration below, OpenAI API key is read from environment variable `OPENAI_API_KEY`.

Figure 12. Set OpenAI API key

```yaml
spring:
  ai:
    openai:
      apiKey: ${OPENAI_API_KEY}
```

Consolidate Local and Production Environment

If we use Ollama for local development and OpenAI for production, we need to have add both model dependencies in Spring AI. These two dependencies will conflict with each other. We should consolidate these two models. We can use only OpenAI model but different API endpoints in development and production.

Many model services provide an API which is compatible with OpenAI. Ollama also has this API. After Ollama is started, this API can be accessed from base URL `http://localhost:11434/v1/`.

⚠ OpenAI compatibility of Ollama is experimental and is subject to major adjustments including breaking changes. Only parts of OpenAI API are supported.

We can use Spring profiles to apply configurations for different environments. For the development profile, `spring.ai.openai.baseUrl` is configured to `http://localhost:11434/v1`. The API key is required for configuration, but will be ignored, so the value can be anything.

Figure 13. Use Ollama OpenAI compatible API in development profile

```yaml
spring:
  ai:
    openai:
      baseUrl: http://localhost:11434/v1
      apiKey: ollama
```

In the production profile `application-prod.yaml`, `spring.ai.openai.baseUrl` is configured to `https://api.openai.com/v1`, which is the endpoint of OpenAI API.

Figure 14. Use OpenAI in production profile

```yaml
spring:
  ai:
    openai:
      baseUrl: https://api.openai.com/v1
      apiKey: ${OPENAI_API_KEY}
```

Profiles can be switched using the option `-Dspring.profiles.active`, e.g. `-Dspring.profiles.active=prod`.

Depends on whether you want to run models locally, there are two recommendations about setup of development environment.

Cloud-based Model Services

Cloud-based model services are actually cheap to use. One option is to simply use model services for both development and production. Spring AI provides integration modules for popular model service platforms. We only need to include the Spring AI module and configure it.

Let's use Anthropic Claude as an example. In a Spring Boot application, we can add the dependency of `spring-ai-starter-model-anthropic` module.

Figure 15. Anthropic module dependency

```xml
<dependency>
  <groupId>org.springframework.ai</groupId>
  <artifactId>spring-ai-starter-model-anthropic</artifactId>
</dependency>
```

Then we can configure Anthropic Claude. The prefix of configuration properties is `spring.ai.anthropic`. An API key is required to be configured as environment variable `ANTHROPIC_API_KEY`. The model `claude-opus-4-0` is used.

Figure 16. Configure Anthropic Claude

```yaml
spring:
  ai:
    anthropic:
      apiKey: ${ANTHROPIC_API_KEY}
      chat:
        options:
          model: claude-opus-4-0
```

Use Container

If you want to run models locally, It's recommended to run models in a container. Container tools like Docker and Podman have already been used extensively in development. You may already use containers to run databases, message brokers, and other tools. Running models in a container means that you don't need to install other tools.

llama.cpp

A popular choice is using llama.cpp to run models. `llama.cpp` provides an OpenAI compatible API to interact with the model. Model files can be downloaded from Hugging Face.

In the Docker compose file below, the model file of `Qwen3-0.6B` is downloaded from Hugging Face, then `llama.cpp` is started to serve this model.

Figure 17. Docker compose file to run models using llama.cpp

```yaml
services:
  model-runner:
    image: ghcr.io/ggml-org/llama.cpp:server
    volumes:
      - model-files:/models
    command:
      - "--host"
      - "0.0.0.0"
      - "--port"
      - "8080"
      - "-n"
      - "512"
      - "-m"
      - "/models/Qwen3-0.6B-Q8_0.gguf"
    ports:
      - "8180:8080"
    depends_on:
      model-downloader:
        condition: service_completed_successfully

  model-downloader:
    image: ghcr.io/alexcheng1982/model-downloader
    restart: "no"
    volumes:
      - model-files:/models
    command:
      - "hf"
      - "download"
      - "unsloth/Qwen3-0.6B-GGUF"
      - "Qwen3-0.6B-Q8_0.gguf"
      - "--local-dir"
      - "/models"

```

```yaml
volumes:
  model-files:
```

After the container is started, the model API can be accessed from `http://localhost:8180`. In Spring AI, we can create a new profile which sets the configuration key `spring.ai.openai.baseUrl` to `http://localhost:8180`. The `apiKey` can be set to anything.

Figure 18. Use OpenAI compatible API running in the container

```yaml
spring:
  ai:
    openai:
      baseUrl: http://localhost:8180
      apiKey: demo
```

ℹ llama.cpp provides a web UI to interact with the model. You can access this UI at `http://localhost:8180` using a browser.

Ollama

Ollama can also run in a container, which means we don't need to install Ollama on local machine.

In the Docker compose file below, Ollama is started in a container. Another container is used to pull the `qwen3:0.6b` model.

Figure 19. Docker compose file to run models using Ollama

```yaml
services:
  ollama:
    image: ollama/ollama
    container_name: ollama
    ports:
      - "11434:11434"
    volumes:
      - ollama:/root/.ollama
    restart: unless-stopped
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localhost:11434"]
      interval: 30s
      timeout: 10s
      retries: 5
    command: ["/bin/ollama", "serve"]

  ollama-pull-qwen3:
    image: ollama/ollama
    container_name: ollama-pull-qwen3
    volumes:
      - ollama:/root/.ollama
    depends_on:
      ollama:
        condition: service_healthy
    command: ["/bin/ollama", "pull", "qwen3:0.6b"]

volumes:
  ollama:
    driver: local
```

Chat Completion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ChatModel

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Message

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ChatOptions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ChatResponseMetadata

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Create Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Chat Response

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ChatClient

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Build Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Call Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Handle Response

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ChatClient Builder

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Use Custom Chat Options

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Global Default Chat Options

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Request Level Chat Options

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Prompt Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

PromptTemplate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Builder

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Custom `TemplateRenderer`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Advisor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`Advisor`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`CallAdvisor`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`StreamAdvisor`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`BaseAdvisor`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Update `ChatClientRequest`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`LoggingAdvisor`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Use `AdvisorSpec`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Enable `Advisors`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Advisors Order

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Recursive Advisor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Streaming Chat Completion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

StreamingChatModel

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Streaming Web Application

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Streaming JSON Events

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Structured Output Converter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

StructuredOutputConverter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ListOutputConverter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MapOutputConverter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

BeanOutputConverter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Use ChatClient

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Multimodal Input

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Media

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Image Understanding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Retrieval-Augmented Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

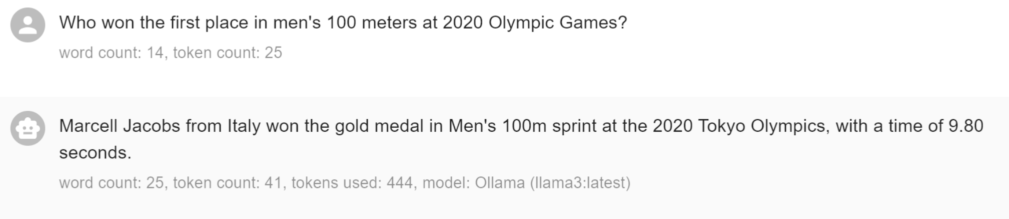
RAG Introduction

In this part, we'll start the journey of building Retrieval-Augmented Generation (RAG) applications using Spring AI. RAG applications have already gained a lot of attentions from industry. It's a typical area where AI technology has practical usage.

Reduce Hallucinations

The main goal of using RAG is to reduce hallucinations when using language models for chat completions. Language models are trained using material from various sources. Once a language model is trained, its parametric knowledge is frozen. When the prompt to a language model contains information not seen in the training materials, the output will hallucinate.

For example, when asking a language model (Llama 3) the following question: *Who won the gold medal in men's 100 meters at 2020 Olympic Games?*. The language model can provide the correct result.



Who won the first place in men's 100 meters at 2020 Olympic Games?

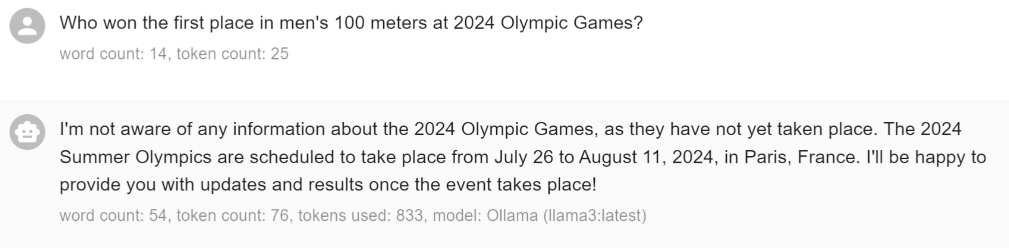
word count: 14, token count: 25

Marcell Jacobs from Italy won the gold medal in Men's 100m sprint at the 2020 Tokyo Olympics, with a time of 9.80 seconds.

word count: 25, token count: 41, tokens used: 444, model: Ollama (llama3:latest)

Figure 20. QA using model internal knowledge

If we change the question to *Who won the gold medal in men's 100 meters at 2024 Olympic Games?*, the language model cannot provide a meaningful answer.



Who won the first place in men's 100 meters at 2024 Olympic Games?

word count: 14, token count: 25

I'm not aware of any information about the 2024 Olympic Games, as they have not yet taken place. The 2024 Summer Olympics are scheduled to take place from July 26 to August 11, 2024, in Paris, France. I'll be happy to provide you with updates and results once the event takes place!

word count: 54, token count: 76, tokens used: 833, model: Ollama (llama3:latest)

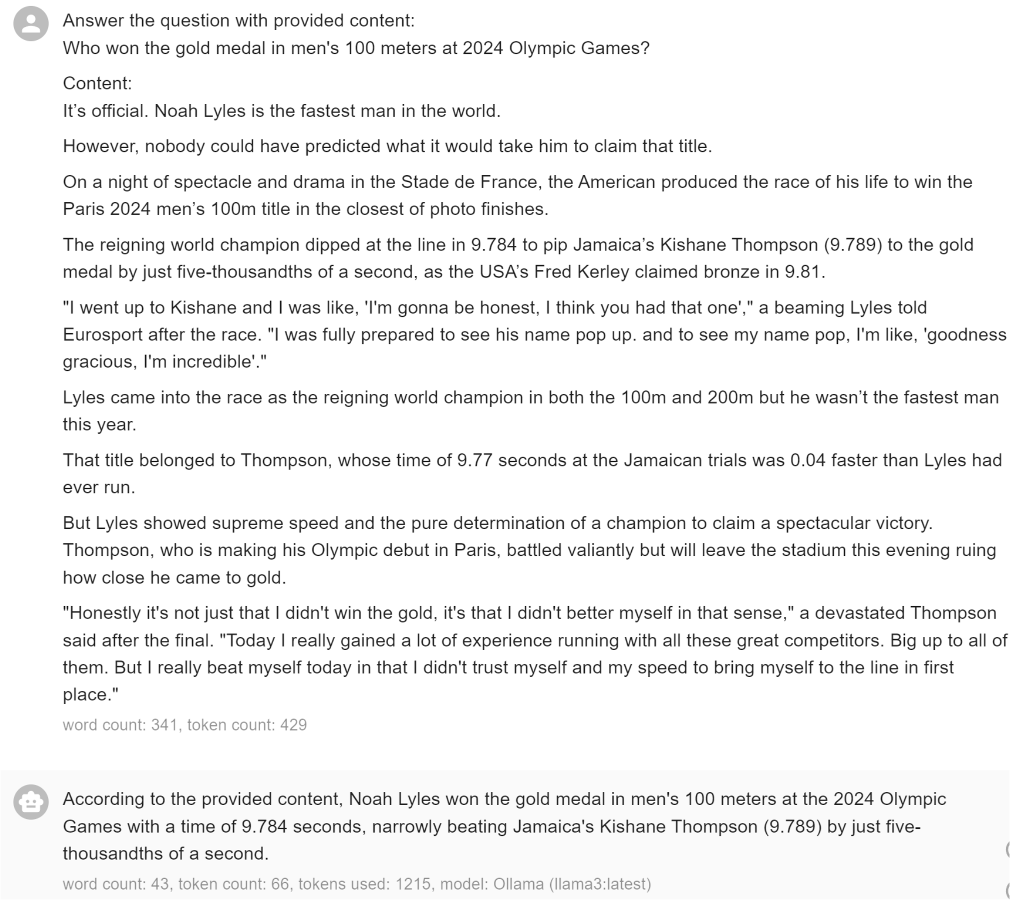
Figure 21. QA without model internal knowledge

There are three approaches to reduce hallucinations, model fine-tuning, tools, and RAG.

- **Model fine-tuning** works by fine-tuning the model to include extra materials.
- **Tools** allow a model to interact with external systems to provide necessary information.
- **RAG** works by augmenting the original prompts with retrieved content for a model to generate output.

The idea behind RAG is quite simple. A frozen language model itself lacks information to generate output for certain prompts. If we augment the prompt to include content from external sources, the model can leverage the provided content to generate the output. The included content is retrieved from an external system and must be semantically similar with the original prompt.

Let's go back to the example above. If we augment the original prompt with the content from a web page, the model can generate meaningful output.



Answer the question with provided content:
Who won the gold medal in men's 100 meters at 2024 Olympic Games?

Content:
It’s official. Noah Lyles is the fastest man in the world.

However, nobody could have predicted what it would take him to claim that title.

On a night of spectacle and drama in the Stade de France, the American produced the race of his life to win the Paris 2024 men’s 100m title in the closest of photo finishes.

The reigning world champion dipped at the line in 9.784 to pip Jamaica’s Kishane Thompson (9.789) to the gold medal by just five-thousandths of a second, as the USA’s Fred Kerley claimed bronze in 9.81.

"I went up to Kishane and I was like, 'I'm gonna be honest, I think you had that one'," a beaming Lyles told Eurosport after the race. "I was fully prepared to see his name pop up. and to see my name pop, I'm like, 'goodness gracious, I'm incredible'."

Lyles came into the race as the reigning world champion in both the 100m and 200m but he wasn’t the fastest man this year.

That title belonged to Thompson, whose time of 9.77 seconds at the Jamaican trials was 0.04 faster than Lyles had ever run.

But Lyles showed supreme speed and the pure determination of a champion to claim a spectacular victory. Thompson, who is making his Olympic debut in Paris, battled valiantly but will leave the stadium this evening ruing how close he came to gold.

"Honestly it's not just that I didn't win the gold, it's that I didn't better myself in that sense," a devastated Thompson said after the final. "Today I really gained a lot of experience running with all these great competitors. Big up to all of them. But I really beat myself today in that I didn't trust myself and my speed to bring myself to the line in first place."

word count: 341, token count: 429

According to the provided content, Noah Lyles won the gold medal in men's 100 meters at the 2024 Olympic Games with a time of 9.784 seconds, narrowly beating Jamaica's Kishane Thompson (9.789) by just five-thousandths of a second.

word count: 43, token count: 66, tokens used: 1215, model: Ollama (llama3:latest)

Figure 22. QA with provided content

Naive RAG

Naive RAG is a simple way to implement RAG. As the name suggests, naive RAG may not provide best results. However, naive RAG is a good starting point to leverage RAG.

The diagram below shows the architecture of naive RAG. Given a user prompt, similar documents related the prompt are retrieved from a vector database. Similar documents and the original prompt are combined together to form the final input to a language model to generate the output.

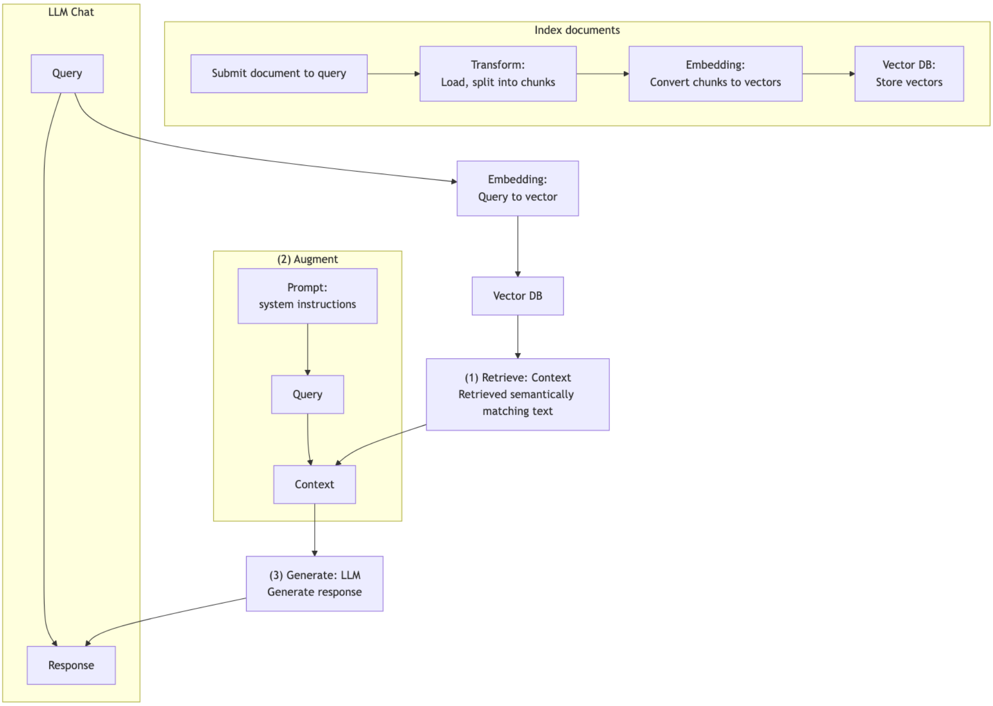


Figure 23. Naive RAG

Vector databases play an important role in RAG. Reference documents are stored in vector databases. Text content is converted to a vector (an array of floating point numbers) using a text embedding model. For a given document, its content and embedding vector are both stored in a vector database. For a prompt, its content is also converted to a vector using the same embedding model. Similarity between the vector of prompt and vectors of reference documents are calculated using vector similarity algorithms. By doing this, similarity between documents is converted to similarity between vectors.

After retrieving similar documents from the vector database, these documents are combined with the original prompt. The combination is usually done with a predefined template. In the template below, we instruct the language model to answer the question using provided content.

Figure 24. Prompt template

```
Answer the question using provided content:

Question: {question}

Provided content: {content}

Answer:
```

Embedding Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`EmbeddingModel`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`EmbeddingOptions`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

`BatchingStrategy`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Use `EmbeddingModel`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Documents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Create Documents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Document Reader

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

TextReader

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

JsonReader

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Document Transformer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Document Writer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Vector Store

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

VectorStore

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Create `VectorStore`

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Add Documents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Delete Documents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Similarity Search

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

VectorStore REST API

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

SimpleVectorStore

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Save and Load

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Pgvector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Installation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Spring AI Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Vector Store Cloud Services

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MongoDB Atlas

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

RAG

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Simple RAG

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Modular RAG

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Query

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Pre-Retrieval

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

QueryTransformer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

RewriteQueryTransformer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

TranslationQueryTransformer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

CompressionQueryTransformer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

QueryExpander

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MultiQueryExpander

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Retrieval

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

DocumentRetriever

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

VectorStoreDocumentRetriever

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

DocumentJoiner

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ConcatenationDocumentJoiner

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Post-Retrieval

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

DocumentSelector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

DocumentRanker

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

DocumentCompressor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

QueryAugmenter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ContextualQueryAugmenter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

RetrievalAugmentationAdvisor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

RAG Examples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Text-to-SQL

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Database Metadata

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Use Advisor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Execute SQL Statements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

REST Controller

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Test

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

PDF Q&A

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Load PDF

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Q&A

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Deployments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Ollama

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

OpenAI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Test

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MCP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MCP Introduction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Quick Start

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Java Development Basics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

stdio Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

HTTP SSE Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MCP Client

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Spring Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Spring WebMVC

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Spring WebFlux

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Client

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Spring AI Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Client

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MCP Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Shared Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

ServerExchange

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Prompt Templates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Resource

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Resource Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Client Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Usage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Completions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Logging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Pagination

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MCP Client

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Roots

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Sampling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

MCP Examples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

File System MCP Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Completions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Agent Introduction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Cooking Suggestion Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Agent Components

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Profile

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Persona

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Knowledge and memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Reasoning and evaluation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Planning and feedback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Agentic Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Task Execution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Scope of Objective

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

LLM Options

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Prompt Template

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Evaluator-Optimizer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Use Different Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Max Number of Evaluations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Evaluation Results

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Generate Initial Result

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Evaluate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Optimize

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Parallelization Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Result Types of Subtasks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Assembling Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Rate Limits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Routing Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Route Selection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Chain Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Agent as Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Tool as Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Orchestrator-Workers Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Example

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Agent Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Persona

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Knowledge

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Food Nutrition Facts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Reasoning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Task

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.

Source Code and Materials

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/spring-ai.