

@agoncal

Understanding LangChain4j

LangChain4j 1.x

2nd Edition

Antonio Goncalves

Foreword by Dmytro Liubarskyi

Understanding LangChain4j

LangChain4j 1.x

Antonio Goncalves

2025-06-16: 2nd Edition

Table of Contents

Foreword	3
About the Author	4
Acknowledgments	6
Introduction	8
Where Does This Fascicle Come From?	8
Who Is This Fascicle For?	9
How Is This Fascicle Structured?	9
Conventions	11
The Sample Application	11
Downloading and Running the Code	13
Getting Help	14
Contacting the Author	14
I: First Steps	15
1. First Look at LangChain4j	16
2. Understanding LangChain4j 1.x	18
2.1. Understanding AI Models	18
2.1.1. Generative AI	18
2.1.2. Large and Small Language Models	19
2.1.3. Training	20
2.1.4. Types of Models	20
2.1.5. Model Providers	21
2.1.6. Prompts	21
2.1.7. Roles and Messages	22
2.1.8. Hallucinations	25
2.1.9. Tokens	26
2.1.10. Context Window	26
2.1.11. Function Calling	27
2.1.12. Use Cases	27
2.2. Understanding Embeddings	28
2.2.1. Dimensions	28
2.2.2. Types of Embedding Models	29
2.2.3. Vector Databases	30
2.2.4. Semantic Search	30
2.3. Understanding Retrieval-Augmented Generation (RAG)	31
2.3.1. Fine Tuning vs RAG	32
2.3.2. Processing In-house Data	32
2.3.3. Splitting Documents	33
2.4. LangChain4j Overview	35

2.4.1. Internal Architecture of LangChain for Java	35
2.4.2. A Brief History of LangChain for Java	36
2.4.3. Other Competing Technologies	36
2.5. Summary	37
3. Getting Started	39
3.1. Developing Your First LangChain4j Application	39
3.1.1. Setting up the Maven Dependencies	40
3.1.2. The Jazz Musician Record	43
3.1.3. Invoking an OpenAI GPT Model	43
3.1.4. Getting an OpenAI Key	45
3.1.5. Executing the Code	45
3.1.6. Checking the Request and the Response	45
3.1.7. Testing the Code	47
3.1.8. Executing the Tests	50
3.2. A Technical Look at LangChain4j	51
3.2.1. Main LangChain4j Source Code Repositories	52
3.2.2. Main LangChain4j Modules	52
3.2.3. Main LangChain4j Packages	55
3.2.4. Main LangChain4j APIs	56
3.2.5. Main LangChain4j Annotations	57
3.2.6. Main LangChain4j External Dependencies	58
3.2.7. LangChain4j Configuration Files	58
3.3. Summary	59
II: AI Models	62
4. Accessing Models	63
4.1. REST APIs vs SDKs	63
4.2. Types of Models Supported by LangChain4j	65
4.2.1. Language Models	67
4.2.2. Chat Models	68
4.2.3. Image Models	70
4.2.4. Moderation Models	72
4.2.5. Scoring Models	74
4.3. Configuring the Model's Request	75
4.3.1. Hallucinations	79
4.4. Handling the Model's Response	81
4.4.1. Typed and Untyped Response	82
4.4.2. JSON Response	83
4.4.3. Streaming the Response	85
4.5. Model Providers Supported by LangChain4j	87
4.5.1. Amazon Bedrock	91
4.5.2. Azure OpenAI	92

4.5.3. GitHub Models	94
4.5.4. Google Vertex AI and Vertex AI Gemini	95
4.5.5. HuggingFace	96
4.5.6. MistralAI	97
4.5.7. Ollama	98
4.5.8. OpenAI	99
4.6. Summary	102
5. Invoking Models	104
5.1. Tokens and Context Window	104
5.1.1. Tokens	104
5.1.2. Context Window	107
5.2. Roles and Messages	108
5.2.1. Roles	108
5.2.2. Message Types	109
5.2.3. Content Types	111
5.2.4. Message Templates	113
5.3. Maintaining a Conversation	115
5.3.1. Sending Previous Messages	116
5.3.2. Chat Memory	117
5.3.3. Memory Store Supported by LangChain4j	121
5.3.4. Chat Memory Store	122
5.4. Summary	125
6. Enriching Models	126
6.1. Tools	126
6.1.1. Understanding Tools	126
6.1.2. Defining Tools	128
6.1.3. Calling Tools	128
6.2. Model Context Protocol	132
6.2.1. Understanding MCP	132
6.2.2. Defining an MCP Server	134
6.2.3. Calling the MCP Server	135
6.2.4. Under the Hood	137
6.3. Summary	140
III: Retrieval-Augmented Generation	142
7. Processing Documents	143
7.1. Documents	143
7.2. Parsing Documents	146
7.3. Loading Documents	148
7.4. Transforming Documents	152
7.5. Splitting Documents into Segments	153
7.6. Summary	156

8. Handling Embeddings	157
8.1. Embeddings	157
8.2. Embedding Models Supported by LangChain4j	160
8.2.1. Remote Embedding Models	160
8.2.2. Local In-Process Embedding Models	163
8.3. Embedding Stores Supported by LangChain4j	165
8.3.1. Azure AI Search	168
8.3.2. Elasticsearch	169
8.3.3. MongoDB	170
8.3.4. PGVector	170
8.3.5. Qdrant	171
8.3.6. In-memory Embedding Store	172
8.4. Manipulating Embeddings	173
8.4.1. Storing Embeddings	173
8.4.2. Removing Embeddings	174
8.4.3. Similarity Search	175
8.5. Summary	180
9. RAG	182
9.1. Limitations of AI Models	182
9.2. Retrieval-Augmented Generation	183
9.3. The Naive RAG Workflow	184
9.4. Implementing RAG with LangChain4j	185
9.5. Summary	189
IV: Simplifying Generative AI	190
10. AI Services	191
10.1. AI Services	191
10.2. Accessing Models	192
10.2.1. Streaming the Response	193
10.3. Using Messages and Templates	195
10.4. Maintaining a Conversation	197
10.5. Moderating Chats	198
10.6. Extending Models with Tools	201
10.6.1. Tools with Parameters	206
10.7. Getting Structured Outputs	210
10.8. Summary	213
11. Easy RAG	214
11.1. Understanding Easy RAG	214
11.2. Applying Easy RAG	215
11.3. Summary	216
V: Wrapping Up	217
12. Putting It All Together	218

12.1. Presenting the Vintage Store Chatbot Application	218
12.2. Setting up the Maven Dependencies	220
12.3. Developing the Ingestion Phase	222
12.3.1. Writing the Document Ingestor	223
12.4. Developing the Query Phase	225
12.4.1. Writing the ChatAssistant Interface	226
12.4.2. Writing the ChatService Class	227
12.4.3. Writing the Tools for Legal Documents	229
12.5. Executing the Application	230
12.5.1. Getting an OpenAI Key	230
12.5.2. Compiling the Code	230
12.5.3. Running the Qdrant Vector Database	230
12.5.4. Executing the Document Ingestor	231
12.5.5. Executing the Chat Assistant	233
12.5.6. Checking the Request and the Response	234
12.6. Summary	238
13. References	239
13.1. LangChain4j Resources	239
13.1.1. Documentation and Code	239
13.1.2. Community and Support	239
13.1.3. Framework Integrations	239
13.2. AI Model Providers	240
Conclusion	241
Appendixes	243
Appendix A: Setting up the Development Environment on macOS 15	244
Homebrew 4.x	244
SDKMAN! 5.x	245
Java 21	246
Maven 3.9.x	249
Docker 28.x	253
Testing Frameworks	262
JUnit 5	262
TestContainers 1.20.x	267
Git 2.x	269
Appendix B: Setting up Model Providers	271
Amazon Bedrock	271
Anthropic	274
Azure OpenAI	277
Cohere	280
DeepSeek	283
GitHub Models	285

Google AI Gemini	287
Hugging Face	289
Mistral AI	291
Ollama	297
OpenAI	301
Appendix C: LangChain4j Versions	305
Appendix D: Revisions of the Fascicle	318
Appendix E: Resources by the Same Author	319
Fascicles	319
Understanding Bean Validation	319
Understanding JPA	320
Understanding LangChain4j	320
Understanding Quarkus	321
Practising Quarkus	322
Online Courses	323
Starting With Quarkus (<i>Udemy</i>)	323
Building Microservices With Quarkus (<i>Udemy</i>)	324
Accessing Relational Databases with Quarkus (<i>Udemy</i>)	325
Quarkus: Fundamentals (<i>PluralSight</i>)	325
Microservices: The Big Picture (<i>PluralSight</i>)	326
Java EE: The Big Picture (<i>PluralSight</i>)	326
Java EE: Getting Started (<i>PluralSight</i>)	326
Java EE 7 Fundamentals (<i>PluralSight</i>)	327
Java Persistence API 2.2 (<i>PluralSight</i>)	327
Contexts and Dependency Injection 1.1 (<i>PluralSight</i>)	328
Bean Validation 1.1 (<i>PluralSight</i>)	328
Appendix F: Printed Back Cover	330

Understanding LangChain4j

Copyright © 2018-2025 by Antonio Goncalves

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "*Attention: Permissions Coordinator*," at the email address below:

agoncal.fascicle@gmail.com

Trademarked names, logos, and images may appear in this fascicle. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image, I use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The distribution of this fascicle is made through Amazon KDP (*Kindle Direct Publishing*^[1]).

Any source code referenced by the author in this text is available to readers at <https://github.com/agoncal/agoncal-fascicle-langchain4j/tree/1.0.1>. This source code is available for reproduction and distribution as it uses an MIT licence^[2].

- www.antoniogoncalves.org
- agoncal.teachable.com
- www.amazon.com/author/agoncal

You can find two different formats of this fascicle:

- eBook (PDF/EPUB) at <https://agoncal.teachable.com/p/ebook-understanding-langchain4j>
- Paper book (ISBN: 9798338389195) and eBooks at <https://www.amazon.com/dp/B0DHXGS1LM>

Understanding LangChain4j

LangChain4j 1.x

2nd Edition

2025-06-16

To my mum, Rita Navalhas.

To my great-cousin, "Soldado" Navalhas.

To my cousin, Carlos Navalhas.

Foreword

Dear Reader,

Welcome to this book on LangChain4j!

I'm Dmytro Liubarskyi, the developer behind LangChain4j. My passion for both Java and artificial intelligence led me to create LangChain4j, a library that connects the worlds of Generative AI and Java.

In the rapidly changing field of GenAI (Generative AI), finding comprehensive and up-to-date resources can be challenging. That's why this book stands out. It not only demonstrates the capabilities of LangChain4j but also provides an in-depth exploration of the latest GenAI concepts.

I'm honoured that Antonio has dedicated his time and expertise to this book. He is an exceptional writer, known for his ability to make complex topics easy to understand. In these pages, he makes GenAI concepts clear and practical for Java developers.

Whether you're new to AI development or already experienced, I'm confident you'll find valuable insights in these pages. I'm excited for you to explore how LangChain4j can help you build AI-enhanced applications.

Dmytro Liubarskyi

Creator and Lead Developer of LangChain4j

[@LangChain4j](#)

[1] KDP <https://kdp.amazon.com>

[2] MIT licence <https://opensource.org/licenses/MIT>

About the Author



I am a Principal Software Engineer at Microsoft living in Paris. Having been focused on Java development since the late 1990s, my career took me to many different countries and companies where I worked as a consultant. As a former employee of BEA Systems (acquired by Oracle), I developed a very early expertise on distributed systems and then microservices. Today, with my role at Microsoft, I help customers to build and run their intelligent Java applications on Azure AI. AI has become my new tool in the last few years.

I am particularly fond of open source, and I am a member of the OSSGTP^[1] (*Open Source Solution Get Together Paris*). I love to create bonds with the community. So, I created the Paris Java User Group^[2] in 2008 and co-created Devovx France^[3] in 2012 and Voxxed Microservices in 2018^[4].

I wrote my first book on Java EE 5^[5], in French, in 2007. I then joined the *Java Community Process* (JCP)^[6] to become an Expert Member of various *Java Specification Requests* (JSRs) (Java EE 8, Java EE 7, Java EE 6, CDI 2.0, JPA 2.0, and EJB 3.1) and wrote *Beginning Java EE 6* and *Beginning Java EE 7* with Apress^[7]. Still hooked on sharing my knowledge, I decided to then self-publish my later fascicles (on *Jakarta Persistence*, *Jakarta Bean Validation*, *Quarkus*, and *LangChain4j*) as well as online video courses (see [Appendix E](#)).

For the last decades, I have given talks at international conferences (JavaOne, Devovx, GeeCon, and many *Java User Groups*), mainly on Java, distributed systems, microservices, cloud computing and artificial intelligence. I also wrote numerous technical papers and articles for IT websites (DevX) and IT magazines (Java Magazine, Programmez, Linux Magazine). Since 2009, I have been part of the French Java podcast called *Les Cast Codeurs*^[8].

In recognition of my expertise and all of my work for the Java community, I was elected **Java Champion**^[9].

I am a graduate of the *Conservatoire National des Arts et Métiers*^[10] (CNAM) in Paris (with an engineering degree in IT), *Brighton University*^[11] (with an MSc in object-oriented design), *Universidad del Pais Vasco*^[12] in Spain, and *UFScar University*^[13] in Brazil (MPhil in Distributed Systems). I also taught for more than 10 years at the Conservatoire National des Arts et Métiers where I previously studied.

Follow me on LinkedIn ([agoncal](#)), BlueSky ([agoncal.bsky.social](#)), X/Twitter ([@agoncal](#)) or on my blog ([www.antonioagoncalves.org](#)).

[1] OSSGTP <https://www.ossctp.org>

[2] Paris JUG <https://www.parisjug.org>

[3] Devovx France <https://devovx.fr>

[4] Voxxed Microservices <https://voxxeddays.com/microservices>

[5] Amazon <https://www.amazon.com/author/agoncal>

[6] JCP <https://jcp.org>

[7] Amazon <https://www.amazon.com/author/agoncal>

[8] Les Cast Codeurs <https://lescastcodeurs.com>