

Generative AI and Java Using Spring AI and Spring Boot

By Dr. Vishwanath Rao

Prerequisites:

Basic knowledge of Java and the Spring Framework
Understanding of fundamental AI and ML concepts
Familiarity with RESTful web services and application deployment

Course Format:

Lectures and readings
Hands-on coding exercises and labs
Quizzes and assessments
Discussion forums and peer review
Real-world project work and case studies

Tools and Technologies:

Spring Boot and Spring AI modules
AI frameworks (e.g., TensorFlow, PyTorch)
Data processing tools and libraries
Cloud services and containerization platforms

COURSE CONTENTS

Introduction to Generative AI

Understanding Generative AI

Definition and scope of Generative AI
Key differences between generative and discriminative models
Applications and real-world examples

Types of Generative Models

Generative Adversarial Networks (GANs): Architecture, training process, and

applications

Variational Autoencoders (VAEs): Concept, latent space, and use cases

Transformers: Role in text generation and large-scale models like GPT

Challenges and Considerations

Ethical implications and biases

Computational requirements and resources

Model evaluation and performance metrics

Working with Generative AI Models in Java

Setting Up the Environment

Installing Java and necessary libraries

Setting up Maven/Gradle for project dependencies

Configuring IDEs for AI development

Implementing Generative Models

Deeplearning4j:

- Building and training GANs

- Implementing VAEs

TensorFlow Java API:

- Loading and using pre-trained models

- Creating custom models for text or image generation

Example Projects:

- Simple GAN for image generation

- VAE for data reconstruction

Working with Pre-trained Models

Loading models from external sources

Integrating with Java applications using APIs

Fine-tuning and customizing pre-trained models

Developing Generative AI Applications

Text Generation Application

Creating a text generation service using transformers

Implementing a chatbot or text-based game

Handling user inputs and generating responses

Image Generation Application

Building an application to generate or modify images using GANs

Integrating with image processing libraries

Example: AI-based art generator

Data Processing and Integration

Handling and preprocessing data for AI models

Integrating data sources and managing data pipelines

Example: Real-time data processing for AI applications

Introduction to Spring AI

Overview of Spring Framework and Spring Boot

Introduction to AI and ML concepts

The role of Spring AI in modern applications

Use cases and benefits of integrating AI with Spring applications

Getting Started with Spring AI

Setting up a Spring Boot project

Adding dependencies for Spring AI

Configuration and basic setup

Exploring Spring AI's architecture and components

Working with AI Models

Understanding different types of AI models (e.g., supervised vs. unsupervised, neural networks, etc.)

How to import and use pre-trained models

Techniques for training custom models

Evaluating and optimizing AI models

Integrating AI with Spring Boot

Loading and managing AI models in a Spring Boot application

Using Spring Boot's configuration management for AI models

Implementing AI model inference and prediction services

Best practices for handling AI model lifecycle and versioning

Building AI Services

Creating RESTful APIs for AI services

Exposing AI functionality through Spring Web services

Handling requests and responses in the context of AI services

Securing AI services with Spring Security

Data Handling and Processing

Data ingestion and preprocessing for AI

Integrating with data sources (e.g., databases, files, streams)
Implementing data pipelines within a Spring Boot application
Ensuring data quality and consistency

Testing and Debugging AI Applications

Unit and integration testing for AI components
Mocking AI models and services for testing purposes
Debugging common issues with AI models and services
Performance tuning and optimization

Deploying AI Applications

Containerizing Spring AI applications with Docker
Deploying to cloud platforms (e.g., AWS, Azure, Google Cloud)
Scaling AI applications and managing resources
Monitoring and logging for AI services

Advanced Topics

Implementing distributed AI models and services
Leveraging Kubernetes for managing AI workloads
Exploring advanced AI frameworks and libraries (e.g., TensorFlow, PyTorch) within Spring
Using Spring Data and Spring Integration for AI-related tasks

Case Studies and Real-world Examples

Case studies of successful AI implementations using Spring
Real-world examples and project walk-throughs
Analyzing common challenges and solutions in AI integration

Future Trends and Emerging Technologies

Overview of emerging trends in AI and how they integrate with Spring
The impact of new AI technologies on Spring applications
Preparing for future advancements and maintaining up-to-date practices