+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 02/06/2026

S'inscrire

Formation Spring AI : Développer des applications IA

2 jours (14 heures)

Présentation

Spring AI est le framework officiel de l'écosystème Spring, conçu pour consommer et intégrer facilement les modèles d'intelligence artificielle générative dans les applications d'entreprise.

L'un des piliers de Spring AI est son objectif de portabilité : il fournit une API unique et unifiée qui s'abstrait des différents fournisseurs (OpenAI, Google Gemini, Mistral AI, ou des modèles locaux avec Ollama). En s'appuyant sur le protocole OpenAI devenu un standard du marché, le framework applique la philosophie « 1 code, N providers ».

La formation Spring AI couvre à la fois les concepts fondamentaux de l'IA générative (tokens, fenêtres de contexte, température) et les fonctionnalités avancées du framework telles que le Tool Calling, la gestion fine de l'historique (ChatMemory) et la génération augmentée par récupération (RAG).

À l'issue de notre formation Spring AI, vous saurez développer des applications Java intelligentes et connectées aux LLM. Vous saurez structurer les dialogues (messages system, user, assistant) et récupérer directement des réponses sous forme d'objets Java typés.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre le fonctionnement d'un LLM (tokens, fenêtre de contexte, température, modèles) et le vocabulaire de l'écosystème IA
- Mettre en place un projet Spring Boot intégrant Spring AI et dialoguer avec un modèle d'IA

- Exploiter la portabilité de Spring AI pour changer de fournisseur (OpenAI, Gemini, Groq, Ollama, Anthropic, Mistral) sans réécrire le code
- Gérer la mémoire de conversation (ChatMemory) et les sessions multi-utilisateurs
- Récupérer des réponses structurées (objets Java/Kotlin) plutôt que du texte brut
- Rédiger des prompts efficaces (prompt engineering) et choisir la bonne technique de prompt
- Donner des capacités avancées à l'IA : accès Internet (grounding), appel de fonctions métier (Tool Calling), génération d'images
- Concevoir une solution de RAG (Retrieval Augmented Generation) pour faire répondre l'IA sur ses propres documents

Public visé

- Développeurs Java/Kotlin, développeurs back-end, architectes logiciels
- Lead techniques souhaitant intégrer des modèles d'IA (LLM) dans leurs applications

Pré-requis

- Connaissance pratique du framework Spring / Spring Boot (injection de dépendances, starters, @RestController/@Controller, application.properties)
- Maîtrise du langage Java ou Kotlin
- Aucune connaissance préalable en Intelligence Artificielle ou en Machine Learning n'est requise
- Poste de travail équipé d'IntelliJ IDEA et d'un accès Internet

Pré-requis logiciels

- IntelliJ IDEA
- JDK 17+
- Gradle & Spring Boot
- Clé API Gemini gratuite

Programme de notre formation Spring AI

[Jour 1 - Matin]

Comprendre l'IA générative et les LLM

- Introduction à l'IA : LLM, RAG, agents
- Anatomie d'un LLM : les tokens (et la facturation au token), la fenêtre de contexte, l'absence de mémoire entre deux appels, la température, la notion de modèle et de "knowledge cutoff"

Présentation de Spring AI

- Spring AI : le framework officiel de l'écosystème Spring pour consommer des modèles d'IA
- L'objectif de portabilité : une API unique pour de nombreux fournisseurs
- Les briques de Spring AI : ChatClient, Prompts & Templates, Structured Output, ChatMemory, Embeddings & Vector Stores, RAG (Advisors), Tool Calling, Image/Audio
- Le protocole OpenAI devenu standard : "1 code, N providers"

[Jour 1 - Après-midi]

Le paysage des fournisseurs d'IA

- Panorama des providers : OpenAI (ChatGPT), Google Gemini, Groq, Ollama (local), Anthropic (Claude), Mistral AI
- Critères de choix : coût, free tier, vitesse, confidentialité, multimodalité
- Protocole compatible OpenAI vs starter natif du provider : avantages et limites
- Les deux starters OpenAI de Spring AI
- TP - HelloWorld : premier appel à une IA via Spring AI
  - Création d'un projet Spring Boot (Web, Thymeleaf, Google GenAI), obtention d'une clé Gemini gratuite, configuration de l'application.properties, premier prompt et expérimentation de la température

Dialoguer avec le modèle : ChatModel vs ChatClient

- ChatModel (API bas niveau) vs ChatClient (surcouche confortable)
- Construction d'un appel : .prompt() / .call() / .content()

La mémoire de conversation (ChatMemory)

- Le LLM est "stateless" : démonstration de l'absence de mémoire
- Les 3 couches de la mémoire : ChatMemoryRepository (OÙ stocker), ChatMemory (QUOI renvoyer), MessageChatMemoryAdvisor (QUAND l'appliquer)
- Stockage en mémoire (InMemory) vs persistant (JDBC)
- Stratégie de fenêtre de messages (MessageWindowChatMemory) et maîtrise des coûts
- TP - Gestion de l'historique
  - Constater l'absence de mémoire, brancher un MessageChatMemoryAdvisor, gérer une conversation par utilisateur via l'identifiant de session HTTP

Métadonnées et accès à Internet

- Exploiter le ChatResponse : usage des tokens, finishReason, modèle utilisé, rate limit.
- Donner accès à Internet au modèle : la notion de "grounding".
- TP - Internet & Métadonnées
  - Activer la recherche Internet dans le prompt et analyser les métadonnées de la réponse

[Jour 2 - Matin]

Structurer le dialogue

- Les rôles : messages system / user / assistant
- Travailler avec des objets : récupérer directement un objet Java/Kotlin (Structured Output avec .entity(), gérer les listes, et les limites (le LLM peut ne pas renvoyer un JSON valide)

L'art du prompt (Prompt Engineering)

- Les 5 questions d'un bon prompt : Rôle, Tâche, Contexte, Contraintes, Format
- La démarche d'affinage itératif
- Les types de prompt : zero/one/few-shot, recherche, créatif, expansion, résumé, template, méta-prompt, itératif, traduction, narratif
- TP - Prompt Engineering : "Prompt Golf"
  - Obtenir un résultat précis avec le prompt le plus court possible
- TP - Prompt Engineering appliqué : réaliser une page de vente
  - Mise en pratique des techniques de prompt sur un cas concret

[Jour 2 - Après-midi]

Tool Calling (Function Calling)

- Principe : fournir au LLM des fonctions métier qu'il peut demander à appeler (météo, base de données, envoi de mail, calcul...)
- Le LLM n'exécute jamais le code : il demande l'appel (finishReason = TOOL_CALLS), Spring AI réalise l'appel et lui renvoie le résultat
- Les annotations @Tool et @ToolParam ; l'importance du nom et de la description
- L'option returnDirect et l'incompatibilité Tool Calling / accès Internet selon les providers
- TP - Tool Calling
  - Exposer une fonction métier et laisser l'IA l'appeler

Génération d'images

- Génération d'images avec Spring AI
- Le modèle image de Gemini ("Nano Banana") : transformation et génération d'images
- TP - Génération / transformation d'image
  - Générer et transformer une image à partir d'un prompt et d'un fichier

RAG : Retrieval Augmented Generation

- Le principe du RAG : faire répondre l'IA sur VOS documents
- Phase d'ingestion (hors-ligne) : découpe en chunks, transformation en vecteurs (embeddings), stockage dans un VectorStore
- Phase d'interrogation : recherche des chunks pertinents par similarité sémantique (top-k, cosinus)
- Les VectorStores : SimpleVectorStore (en mémoire), PGVector, Redis, Chroma, Milvus...
- Mise en œuvre avec Spring AI : RagConfig, IngestionService, QuestionAnswerAdvisor
- TP - RAG
  - Alimenter une IA avec des documents (ingestion) et l'interroger sur leur contenu

Pour aller plus loin

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d’auto-évaluation nous permettant d’apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d’anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l’acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.

Organisme de formation enregistré sous le numéro 11 75 54743 75. Cet enregistrement ne vaut pas agrément de l’État.
© Ambient IT 2015-2026. Tous droits réservés. Paris, France - Suisse - Belgique - Luxembourg