AZ

PROGRAMME DE FORMATION

IA

Intégrer l'IA dans ses applications : API, agents et RAG

Construisez des applications IA fiables avec API, agents et RAG.

Utiliser ChatGPT est simple. Construire une fonctionnalité IA fiable est un vrai métier d'intégration.

Alexandre Zebboudj

Formateur IT, IA & cybersécurité

zebboudjalexandre@gmail.com

06 51 64 53 63

zebboudj.fr

DURÉE

5 jours (35 heures)

PUBLIC

Développeurs, ingénieurs logiciels, architectes, lead developers, CTO

NIVEAU D'ENTRÉE

AVANCÉ

Avancé - Développeurs, usage régulier d'un langage requise

TARIF ANIMATION

350 €/j

Toute la France Basé en Île-de-France

Intra | Inter | Exter | Présentiel | Distanciel | Hybride

Pourquoi cette formation

Utiliser ChatGPT est à la portée de tous ; intégrer l'IA générative dans un produit ou un système d'information est un métier. Les entreprises qui veulent dépasser le stade des usages individuels ont besoin de développeurs capables de construire des fonctionnalités IA fiables : appels aux API des grands modèles, génération structurée, appels de fonctions, recherche augmentée par génération sur les documents de l'entreprise, et agents capables d'enchaîner des actions. Cette formation intensive de 5 jours forme des développeurs opérationnels sur toute la chaîne : des premiers appels d'API jusqu'au déploiement d'une application RAG complète et d'un agent outillé, en passant par les questions que les tutoriels ignorent : coûts, latence, évaluation de la qualité, sécurité et conformité. Un projet fil rouge est construit sur les 5 jours et chaque participant repart avec une application fonctionnelle et un socle de code réutilisable.

CHIFFRE CLÉ

1 projet

application RAG et agent outillé construits sur les 5 jours

Repères

API LLM | Function calling | RAG | Agents IA | Dépôt de code inclus

Objectifs pédagogiques

- Expliquer le fonctionnement des grands modèles de langage du point de vue de l'intégrateur : tokens, fenêtre de contexte, température, coûts et limites
- Intégrer les API des principaux fournisseurs dans une application : authentification, appels, streaming, gestion des erreurs et quotas
- Concevoir des prompts systèmes robustes et obtenir des sorties structurées fiables en JSON, exploitables par du code
- Mettre en œuvre les appels de fonctions pour connecter un modèle aux données et services de l'entreprise
- Construire un pipeline RAG complet : extraction, découpage, embeddings, base vectorielle, recherche et génération avec citations
- Développer un agent IA : boucle de raisonnement et d'action, outils, garde-fous et supervision
- Évaluer la qualité d'un système IA : jeux de tests, métriques et détection des régressions
- Sécuriser et industrialiser : injection de prompt, protection des données, RGPD, AI Act, coûts, journalisation et mise en production

Prérequis

Usage régulier d'un langage de programmation. Python est recommandé pour les exercices techniques ; les concepts sont transposables à JavaScript, TypeScript et autres langages. Être à l'aise avec les notions d'API REST et de JSON. Aucune connaissance préalable en machine learning n'est exigée.

Programme détaillé

Jour 1 · Les fondations : comprendre et appeler les LLM

01

Les LLM du point de vue du développeur — 2h00

- Fonctionnement d'un modèle de langage sans mathématiques inutiles : prédiction de tokens, fenêtre de contexte, température, top_p et max_tokens
- Panorama des fournisseurs et modèles : OpenAI, Anthropic, Mistral, modèles open source ; critères de choix qualité, coût, latence, confidentialité et hébergement européen
- L'économie d'un projet LLM : tarification par token, estimation et maîtrise des coûts dès la conception
- Présentation du projet fil rouge : un assistant documentaire métier complet, enrichi chaque jour

02

Premiers appels d'API — 2h30

- Anatomie d'un appel : messages, rôles system, user et assistant, paramètres d'inférence
- TP : premiers appels aux API OpenAI et Anthropic, comparaison des réponses et des SDK
- Gérer la conversation : historique, fenêtre de contexte, troncature et résumé
- Streaming des réponses : améliorer l'expérience utilisateur
- Gestion des erreurs, des quotas, des rate limits, des reprises, retries et backoff

03

Prompts systèmes et sorties structurées — 2h30

- Concevoir un prompt système robuste : rôle, consignes, périmètre et refus
- Obtenir du JSON fiable : sorties structurées, schémas, validation et stratégies de réparation
- TP fil rouge : mise en place du socle de l'application, appel API, prompt système et sortie structurée validée
- Revue de code croisée et bons réflexes d'organisation du code IA

Jour 2 · Connecter le modèle au monde réel : function calling

04

Les appels de fonctions — 3h00

- Principe : le modèle décide d'appeler des fonctions que votre code exécute
- Déclarer des outils : schémas, descriptions efficaces et choix des granularités
- La boucle d'exécution : appel, résultat et poursuite du raisonnement
- TP : connecter le modèle à des fonctions métier, recherche dans une base, appel d'une API externe et calculs

05

Cas d'application d'intégration — 3h00

- TP fil rouge : ajout d'outils à l'assistant, consultation de données structurées de l'entreprise fictive et actions simples
- Gérer les cas dégradés : fonction en erreur, réponse hors périmètre et boucles infinies
- Multi-fournisseurs : abstraire ses appels pour ne pas dépendre d'un seul acteur

06

Latence, coûts et modèles adaptés — 1h00

- Choisir le bon modèle pour chaque tâche : routage simple ou complexe, modèles rapides vs modèles puissants
- Mise en cache, traitement par lots et optimisation des prompts
- Atelier : audit de coût et de latence du fil rouge, optimisations mesurées

Jour 3 · RAG : brancher l'IA sur les documents de l'entreprise

07

Comprendre le RAG — 1h30

- Pourquoi le RAG : limites de la connaissance du modèle, confidentialité et fraîcheur des données
- L'architecture complète : ingestion, découpage, embeddings, base vectorielle, recherche et génération
- Les embeddings expliqués simplement : représenter le sens pour rechercher par similarité

08

Construire le pipeline d'ingestion — 2h30

- Extraire le texte des documents réels : PDF, Word, HTML, pièges et solutions
- Le découpage : tailles, chevauchement, découpage sémantique et métadonnées
- TP fil rouge : pipeline d'ingestion du corpus documentaire fourni, vectorisation et stockage dans une base vectorielle

09

Recherche et génération avec citations — 3h00

- La recherche : similarité vectorielle, filtres par métadonnées, recherche hybride mots-clés + vecteurs, reclassement et reranking
- La génération ancrée : construire le prompt avec les extraits, exiger les citations des sources, gérer le je ne sais pas
- TP fil rouge : l'assistant répond désormais à partir des documents, avec citations vérifiables
- Les échecs classiques du RAG et leurs remèdes : mauvais découpage, question mal formulée, contexte noyé

Jour 4 · Agents, évaluation et sécurité

10

Construire un agent

3h00

- Du chatbot à l'agent : boucle de raisonnement et d'action, planification et mémoire de travail
- Concevoir les outils d'un agent et ses garde-fous : périmètre d'action, validation humaine des actions sensibles et limites d'itérations
- TP fil rouge : transformation de l'assistant en agent capable d'enchaîner recherche documentaire, analyse et production d'un livrable structuré
- Quand éviter l'agent : coût, imprévisibilité et alternatives plus simples

11

Évaluer la qualité

2h00

- Pourquoi les tests classiques ne suffisent pas : sorties non déterministes
- Construire un jeu d'évaluation : cas représentatifs, réponses attendues et critères
- Méthodes d'évaluation : règles automatiques, notation par modèle, revue humaine ciblée
- TP : mise en place d'un jeu d'évaluation sur le fil rouge, mesure avant et après une modification de prompt

12

Sécuriser son application LLM

2h00

- L'injection de prompt : démonstrations sur l'application fil rouge, directe et indirecte via les documents ingérés
- Défenses en profondeur : séparation des instructions et des données, filtrage, moindre privilège des outils, validation des sorties
- Protection des données : RGPD, données personnelles dans les prompts et les corpus, choix d'hébergement, conservation, obligations AI Act côté intégrateur
- Journalisation et traçabilité des interactions

Jour 5 · Industrialisation et projet évalué

13

Mettre en production

2h00

- Architecture de référence : API backend, gestion des clés et secrets, files d'attente et supervision
- Suivi en production : coûts, latence, taux d'erreur et dérive de qualité
- Gérer les évolutions des modèles : versions, tests de non-régression et plan de repli
- Revue finale du socle de code réutilisable remis aux participants

14

Projet d'évaluation

3h30

- Cahier des charges remis aux participants : construire, seuls ou en binômes, une fonctionnalité IA complète sur un cas nouveau
- Le projet intègre au minimum sorties structurées et function calling ou RAG, avec jeu d'évaluation minimal et mesures de sécurité
- Développement en autonomie accompagnée

15

Soutenances techniques et clôture

1h30

- Démonstration et revue de code de chaque projet, 15 minutes par participant ou binôme, évaluées sur grille critériée
- Retours détaillés du formateur et axes de progression individuels
- Ressources pour continuer : veille, communautés, certifications, plan d'action personnel et clôture

Moyens pédagogiques

- Exercices guidés et mises en situation, projet fil rouge construit sur les 5 jours, live coding commenté et revues de code croisées
- Dépôt Git fourni avec squelettes, corrections et socle de code réutilisable
- Environnement de travail préparé : guide d'installation transmis en amont, dépendances, clés d'API fournies pour la durée de la formation

Ce avec quoi vous repartez

- Support de formation PDF
- Dépôt de code complet : exemples, squelettes, corrections et socle réutilisable
- Grille de comparaison des fournisseurs et modèles, check-list sécurité des applications LLM, gabarit de jeu d'évaluation, calculateur de coûts

Et concrètement, chaque participant :

- Une application RAG et agent fonctionnelle construite pendant la formation
- Un socle de code réutilisable avec exemples, squelettes et corrections
- Une approche complète : coût, latence, évaluation, sécurité, conformité et production
- Des méthodes transférables à OpenAI, Anthropic, Mistral et aux architectures multi-fournisseurs

Évaluation

- Évaluation diagnostique en début de formation (quiz de positionnement technique)
- Évaluations formatives continues : exercices techniques quotidiens, projet fil rouge et revues de code croisées
- Évaluation sommative au jour 5 : projet technique individuel ou en binôme avec démonstration et revue de code, évalué selon une grille critériée (seuil de réussite défini en amont)
- Questionnaire de satisfaction à chaud et évaluation à froid proposée à 3 mois
- Attestation de fin de formation mentionnant les objectifs atteints, remise à chaque participant
- Chaque participant repart avec une application RAG/agent fonctionnelle, le dépôt de code complet et un jeu d'évaluation prêt à adapter

Modalités d'organisation

- Présentiel ou classe virtuelle ; 5 jours consécutifs ou format 3 + 2 espacés d'une à deux semaines
- 4 à 8 participants
- Formation accessible aux personnes en situation de handicap. Nous contacter en amont afin d'étudier les adaptations possibles (référent handicap de l'organisme de formation).
- Variable selon les modalités de financement et le planning ; en moyenne 2 à 4 semaines après validation de la convention de formation.