Durée :	2 jours soit 14 heures	Référence :	IF-IADEV4

Public visé :

Développeurs confirmés, tech leads, architectes et équipes outillage ou platform engineering qui veulent connecter les agents de code aux systèmes internes et automatiser des chaînes de travail complètes.

Pré-requis :

- Expérience pratique d'un agent de code (niveau IF-IADEV2 ou équivalent).
- Pratique courante de Python ou de TypeScript.
- Notions d'API REST et du format JSON.

Objectifs pédagogiques :

- Développer, tester et sécuriser un serveur MCP qui expose un système interne aux agents.
- Concevoir des skills et des agents personnalisés réutilisables entre outils.
- Orchestrer plusieurs agents spécialisés sur une tâche de développement complexe.
- Exécuter un agent en mode non interactif dans une automatisation.

Compétences acquises à l'issue de la formation :

- Être capable de développer, tester et sécuriser un serveur MCP exposant des outils et des ressources métier.
- Savoir concevoir des skills et des agents personnalisés portables entre les outils compatibles.
- Être en mesure d'orchestrer plusieurs agents spécialisés sur une tâche complexe et d'en contrôler le résultat.
- Savoir automatiser un agent en mode non interactif via sa ligne de commande ou son SDK.

Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure. La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz. L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration. En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.

Moyens et supports pédagogiques :

Cadre présentiel

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique

Cadre téléprésentiel

Session dispensée via notre solution iClassRoom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant

- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert…).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquise ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin

Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.

Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :
Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr
Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.

Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- AGEFIPH
- FIPHFP
- MDPH de votre département

Description / Contenu

Module 1 – Architecture du Model Context Protocol

- Hôte, client, serveur ; outils, ressources, prompts
- Transports : stdio en local, Streamable HTTP à distance
- Versions de la spécification et compatibilité avec les clients du marché
- Démonstration : observer les échanges entre un agent et un serveur avec MCP Inspector

Module 2 – Développer un serveur MCP

- SDK officiels Python ou TypeScript

- Concevoir des outils utilisables par un agent : nom, description, schémas d'entrée et de sortie, erreurs explicites
- Exposer des ressources et des prompts
- Tester : MCP Inspector, tests automatisés, essai dans plusieurs clients
- TP : développer un serveur MCP qui expose une API interne fournie (référentiel de tickets ou catalogue de services) et l'utiliser depuis deux agents différents

Module 3 – Sécuriser et déployer un serveur MCP

- Authentification et autorisation OAuth pour les serveurs distants
- Moindre privilège, validation des entrées, confirmation des actions d'écriture
- Risques : injection d'instructions via les données retournées, descriptions d'outils piégées, fuite de données
- Déploiement en conteneur, journalisation, liste interne des serveurs autorisés
- Atelier : durcir le serveur du TP (lecture seule par défaut, confirmation des écritures, journalisation)

Module 4 – Skills et agents personnalisés

- Structure d'une skill : fichier SKILL.md, scripts, références, chargement à la demande
- Portabilité entre outils (.github/skills, .claude/skills, .agents/skills)
- Agents personnalisés : rôle, outils autorisés, modèle
- Choisir entre instruction, skill, sous-agent et serveur MCP
- TP : créer une skill de génération conforme aux conventions de l'équipe et un agent personnalisé de revue de sécurité

Module 5 – Orchestration multi-agents et automatisation

- Patterns : orchestrateur et sous-agents, chaîne spécification, implémentation, tests, revue, exécution en parallèle
- Isolation du travail par worktrees et branches, fusion et arbitrage
- Coûts, latence, boucles et échecs : détection et reprise
- Exécution non interactive : mode headless des agents en ligne de commande, SDK d'agents (Claude Agent SDK)
- Cas de synthèse évalué : orchestrer une chaîne d'agents qui traite une évolution de bout en bout avec le serveur MCP et les skills créés, validation humaine finale