

# Jenkins, mettre en place l'intégration continue en Java

Cours Pratique de 2 jours - 14h

Réf : JEN - Prix 2025 : 1 930 HT

Le prix pour les dates de sessions 2026 pourra être révisé

Jenkins (anciennement Hudson) est un serveur d'intégration continue permettant d'automatiser les tests, l'audit et déploiement du code réalisé et de restituer des indicateurs qualité sur la production. Ce cours vous présentera toutes les capacités de ce serveur, soutien indispensable au développement en mode Agile.

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Comprendre les principes de l'intégration continue en vue de son implémentation

Intégrer Jenkins avec les autres outils (SCM, gestionnaire de tickets...)

Mettre en place un serveur Jenkins automatisant les build

Automatiser les tests, les audits de code et les déploiements sur la plateforme d'intégration Jenkins

## MÉTHODES PÉDAGOGIQUES

Alternance de théorique et pratique sur l'outil Jenkins.

## LE PROGRAMME

dernière mise à jour : 03/2024

### 1) Introduction à l'intégration continue

- Principes de l'intégration continue.
- Développement agile et intégration continue.
- Prérequis à l'intégration continue.
- Techniques de développement adaptées à l'intégration continue.
- Différents types de tests et d'audit d'une application.
- Cycle de vie du processus de développement.
- Outils annexes et intégration.

*Démonstration : Découverte de Jenkins.*

### 2) Mise en place et automatisation du Build

- Mise en place du serveur, les différents types d'installation.
- Configuration : page principale de la configuration, configuration de Git/SVN, serveur de mail.
- Jenkins et le Build, les meilleures pratiques et méthodes recommandées.
- Stratégies et techniques de notification.
- Fixer les dépendances entre les travaux de Build.
- Jenkins et Maven : rappel sur Maven, configuration du Build Maven, déploiement dans un repository Maven.

*Exercice : Création et configuration d'un Projet Maven Jenkins basé sur un repo Git.*

### 3) Qualité du code

- Introduction, intégration de la qualité dans le processus de build.
- Outils d'analyse : Checkstyle, FindBugs, CPD/PMD.

## PARTICIPANTS

Développeurs, chefs de projet.

## PRÉREQUIS

Connaissance du langage Java et notions du cycle de développement.

## COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

## MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr).

- Configuration du rapport qualité avec le plugin Violations.
- Rapport de complexité, sur les tâches ouvertes.

*Exercice : Adaptation du projet Jenkins pour inclure les outils d'analyse de qualité de code et suivi de la qualité dans le temps.*

#### 4) Automatisation des tests

- Introduction, terminologie du test.
- Automatisation des tests unitaires et d'intégration.
- Configuration des rapports.
- Mesurer la couverture de test.
- Automatisation des tests d'acceptance.
- Automatisation des tests de performance avec JMeter.
- Optimiser les temps d'exécution des tests.

*Exercice : Adaptation du projet Jenkins pour inclure JUnit et suivi des tests de performance avec JMeter.*

#### 5) Automatisation du déploiement

- Mise en place du script de déploiement.
- Mise à jour des bases de données.
- Tests minimaux. Retour en arrière.

*Exercice : Adaptation du projet Jenkins pour automatiser le déploiement de l'artefact construit.*

#### 6) Administration d'un serveur Jenkins

- Activation de la sécurité et mise en place simple.
- Différents types de bases utilisateurs.
- Gestion des autorisations et des rôles.
- Journalisation des actions utilisateur.
- Gestion de l'espace disque.
- Monitoring de la charge CPU.
- Sauvegarde de la configuration.

## LES DATES

---

### CLASSE À DISTANCE

2025 : 09 oct., 18 déc.

2026 : 23 mars, 11 juin, 21 sept.,  
30 nov.

### LILLE

2025 : 18 déc.

### PARIS

2025 : 11 déc.

2026 : 23 mars, 11 juin, 21 sept.,  
30 nov.