

# Formation Intégration Continue avec Maven, Jenkins, Sonar et Git

## Informations

Durée : 4 jours (28h.)

Tarif\* : 2390 € HT

Réf : INTEC

Niveau : Facile

inter intra à distance

\*tarif valable jusqu'au 16/07/2024

## Prochaines sessions

10 juin - 13 juin

()

14 octobre - 17 octobre

()

## Pré-requis

- Avoir des connaissances de base en développement logiciel et/ou gestion de projet

## Objectifs

- Maîtriser les 7 phases de mise en œuvre d'une intégration continue
- Mettre les capacités de Maven en œuvre dans vos applications
- Générer des rapports liés au cycle de vos intégrations
- Comprendre l'acceptance et le déploiement en continu

## Programme

### Introduction à l'Intégration Continue

Comprendre la notion de chaîne d'intégration continue  
Les 7 étapes de mise en œuvre de l'intégration continue  
Le rôle du serveur d'intégration continue  
La place des tests unitaires / intégration  
L'acceptance en continu  
Le déploiement en continu  
L'intégration continue et Agilité  
La robustesse, la maintenabilité

### Maven

Les bases Maven

- Définition
- Conventions
- Coordonnées
- POM
- Installation
- Sections de pom.xml
- Création Projet maven
- Plugin, goal, Mojo
- Phases Maven
- Dépendances
- Repository local, distant
- Plugin help
- Plugin m2eclipse
- Plugin checkstyle
- Plugin sonar

L'architecture de projet Maven

- Le projet multi-modules
- L'héritage, composition
- La génération de rapports
- Le site internet du projet : site
- La gestion de conflits
- Les optimisations pom.xml
- DependencyManagement
- PluginManagement
- Plugin release
- Snapshot

Mise en pratique de Maven

### Nexus

Définition et valeur ajoutée d'un Repository d'entreprise  
L'offre du marché  
L'installation Nexus  
Les fonctionnalités principales de Nexus  
L'intégration Maven / Nexus  
La gestion des livraisons releases / snapshot

# Formation Intégration Continue avec Maven, Jenkins, Sonar et Git

La balise mirror et server dans settings.xml

La balise distributionManagement dans le pom.xml

## Tests

---

Les tests unitaires avec JUnit

Les tests unitaires avec Mockito

Automatiser les tests unitaires

Automatiser les tests d'intégration

L'intégration de tests Selenium WebDriver

Ignorer les tests

La couverture de code

Les tests d'acceptance

Résoudre les problèmes de lenteur des tests

Mise en pratique de tests unitaires avec JUnit

## Git

---

Décentraliser

Créer un dépôt local

Workflow Git

Ajout et versioning de fichiers

Obtenir des informations sur l'état de la copie de travail

Supprimer, déplacer et renommer des fichiers

Analyser l'historique des révisions

Ignorer des fichiers

Créer et gérer des branches

merge : gérer les conflits

Push, pull, fetch

Comprendre rebase, reset

Debugger avec bisect

Gestions des dépôts avec GitHub

Gestions des dépôts avec GitLab

Mise en pratique de Git

## Jenkins

---

Jenkins/Hudson – Historique

- *Préparer l'environnement*
- *Installation rapide*
- *Configuration rapide*
- *Création premier Job*
- *Lancement premier Job*
- *Rapport Javadoc*
- *Rapport couverture de code*
- Premiers pas avec Jenkins

Installation professionnelle de Jenkins

- *Téléchargement et installation*
- *Configuration du serveur*
- *Installer Jenkins sous Linux*
- *Installer Jenkins sous Windows*
- *Mémoire nécessaire*
- *répertoire USER\_HOME/.jenkins*
- *Backup données Jenkins*
- *Upgrade Jenkins*

Configuration professionnelle

- *Tableau de bord de la configuration Jenkins*
- *Configuration Variables d'environnement Système*

# Formation Intégration Continue avec Maven, Jenkins, Sonar et Git

- *Configuration Propriétés globales*
- *Configuration outils de build : Maven, Ant, Shell*
- *Configuration gestionnaire de source (Subversion, CVS, git)*
- *Configuration du serveur de mail*
- *Configuration du serveur proxy*

## Les Jobs dans Jenkins

- *Notion de Job Jenkins*
- *Création d'un job 'freestyle'*
- *Configuration gestionnaire de source*
- *Les triggers*
- *Les étapes*
- *Actions post-build*
- *Lancement de build Maven*
- *Voir les résultats d'un job*

## Les tests dans Jenkins

- *Configurer les rapports de test dans Jenkins*
- *Afficher les résultats de test dans Jenkins*

## Sécurité avec Jenkins

- *Authentification & Autorisation*
- *Activer la sécurité Jenkins*
- *Sécurité simple*
- *Realms / Autorisations*
- *Audit de sécurité Jenkins*

## Les notifications

- *Introduction*
- *Notification mail*
- *Notification RSS*
- *Notification outils de communication*

## La qualité du code

- *La qualité de code dans le processus de build*
- *Outils populaires : PMD, Checkstyle, FindBugs*
- *Rapport outils qualité de code : plugin 'Violations'*
- *Rapport sur la complexité du code*
- *Rapport sur les tâches ouvertes (FIXME, TODO, )*
- *Intégration avec l'outil Sonar*
- *Mise en pratique de Jenkins*

## Sonar

### Les 7 axes de qualité logicielle

L'installation par défaut (base h2)

L'intégration avec MySQL

L'analyse de code via SonarQube Scanner

L'analyse de code via Maven

L'analyse de code via Ant

L'analyse de code via SonarLint dans Eclipse

Le tableau de bord projet

Les différentes métriques de SonarQube

- *L'analyse de bugs potentiels*
- *L'analyse des vulnérabilités*
- *Les respects des règles de codage*
- *L'analyse des résultats de Tests*

# Formation Intégration Continue avec Maven, Jenkins, Sonar et Git

- *L'analyse des duplications de code*
- *L'analyse des commentaires*
- *L'analyse de la complexité du code*

Le paramétrage de règles spécifiques

Les profils de règle

Mise en pratique de SonarQube