

# Développer avec le système de modules de Java



En 2009, la mort du classpath a été annoncée. Il devait être remplacé par un système modulaire et tous nos problèmes de dépendance et de sécurité devaient se résoudre d'eux-mêmes. Le projet Jigsaw est finalement intégré au JDK 9, en septembre 2017.

La modularité se retrouve à la fois dans le JDK et nos applications. Dans cette formation, nous expliquons comment la modularité se retrouve dans le JDK et nos applications. Nous voyons quel problèmes ça résout et quels nouveaux problème ça pose.

Il faut concevoir différemment les dépendances et l'encapsulation, préparons-nous.

2 jours

## Tarif

- intra : 2440 euros HT  
(maxi 6 participants)
- inter : 1080 euros HT

---

## Introduction

- ▶ Classpath et classloader
- ▶ Risque de classes hybrides
- ▶ Défauts de sécurité
- ▶ Problème d'obésité

## Modularité du JDK

- ▶ De rt.jar aux modules
- ▶ Modules standards et module de base
- ▶ Modules dépréciés
- ▶ Modules non-standards
- ▶ Classes et packages supprimés
- ▶ Compilation, packaging et exécution

## Modularité des applications

- ▶ Dépendances entre modules
- ▶ Export de packages
- ▶ Application multi-modules

## Encapsulation des modules

- ▶ Évolution de la visibilité public
- ▶ Règles de répartition de packages
- ▶ Export de packages : globaux et limités
- ▶ Deep reflection
- ▶ Ouverture de package et de module

## Dépendances entre modules

- ▶ Dépendances transitives
- ▶ Dépendances statiques
- ▶ Modules automatiques
- ▶ Mode mixte : module path / class path

## Migration d'applications

- ▶ Option d'accès illégal
- ▶ Ajout de modules racine
- ▶ Ajout de dépendances
- ▶ Évaluation préalable des dépendances

#### Native

- ▶ Construction d'image personnalisée
- ▶ Compilation native