



zenika  
ARCHITECTURE INFORMATIQUE

www.zenika.com

Repartez avec le livre



Durée  
2 jours

Répartition  
40% théorie  
60% pratique

Pré-requis  
Notions de Java

Public  
Architecte  
Développeur  
Chef de projet

Tarif (déjeuners inclus)  
1200 € (HT)  
Tarif (1 mois avant)  
1100 € (HT)

Lieu  
Paris 9ème

Sessions 2009  
26 au 27 janvier  
16 au 17 mars  
18 au 19 mai  
20 au 21 juillet  
14 au 15 septembre  
16 au 17 novembre

Intra-entreprise sur  
demande

Inscription et  
renseignements  
+33(0)1.45.26.19.15  
training@zenika.com  
www.zenika.com

# XML

## Développement XML avec Java

### Contexte

XML peut s'adapter à tout contexte et besoin (gérer des flux RSS, effectuer des appels entre systèmes distants, configurer un système applicatif...). Malgré son omniprésence, XML est souvent source de mises en œuvre inappropriées et coûteuses. Derrière la simplicité relative de XML se cache une syntaxe extrêmement riche et puissante ainsi qu'un ensemble de technologies plus difficiles à appréhender. Nous utiliserons le langage Java dont la puissance des API offre des possibilités variées pour manipuler, analyser, valier et transformer le XML.

### Objectifs

- Connaître les principales API Java permettant de manipuler du XML
- Savoir naviguer, modifier, transformer un document XML en utilisant SAX ou DOM
- Savoir transformer un document XML via XSLT et XPath
- Pouvoir mettre en oeuvre un mapping Java/XML via l'API JAXB

### Contenu

#### XML : eXtended Markup Language

- XML : un langage de balisage
- L'histoire de XML
- Structure et syntaxe d'un document XML
  - balises, commentaires, caractères spéciaux...
- Créer un document XML bien-formé (well-formed XML document)
- Valider un document XML
- Parser un document ou l'analyse syntaxique
- Transformer un document

#### L'écosystème XML

- XML : la première pierre
- XML et sa suite de standards, de modèles et de spécifications
- Un format universel : Web Services, XHTML, OpenDocument...

#### DTD à XSD : décrire et valider un document XML

- Pourquoi définir un modèle ?
- DTD : Document Type Definition
- XSD : XML Schema Description
- Définir un modèle XML : DTD ou XSD ?
- Générer un document XML à partir d'un modèle
- Valider un document XML

#### Les espaces de nommages (ie. namespaces)

- Namespaces : lever les ambiguïtés
- Les namespaces dans XML
- Nommer les éléments et les attributs
- Maîtriser la portée d'un namespace
- Les namespaces dans XSD
- Respecter les conventions de nommage

#### Java et XML : État des lieux

- XML : un langage difficile à manipuler
- Pourquoi utiliser Java pour manipuler XML ?
- JAXP : API standard Java de manipulation XML
- Aperçu de l'API JAXP : valider, parser, transformer...
- Les implémentations JAXP : laquelle choisir ?

#### SAX : lire et traiter un document XML

- SAX : Simple API for XML
- Parcourir un document XML
- Gestion des événements

#### DOM : manipuler un arbre XML

- DOM : Document Object Model
- XML : un arbre de données
- Créer un objet DOM
- Manipuler/Modifier un objet DOM
- Sauvegarder un objet DOM : fichier, base de données...
- DOM comparé à SAX

#### Les alternatives à DOM et SAX

- Les limites de DOM et de SAX
- JDOM n'est pas DOM
- JDOM : s'abstraire totalement du XML
- STAX : Streaming API for XML
- STAX : une démarche hybride entre SAX et DOM

#### XPath : les 'adresses' XML

- La famille XSL
- XPath : pouvoir désigner des éléments dans un document XML
- Les éléments identifiables
- La syntaxe XPath
- Les expressions XPath
- Les fonctions XPath

#### XSLT : transformer un document XML en tout autre document

- XSLT : eXtensible Stylesheet Language Transformations
- XSLT : transformer un modèle XML
- La syntaxe XSLT
- Répétitions, tris et conditions
- XPath et XSLT
- Générer des documents XML/PDF/XHTML à partir d'une même source XML

#### JAXB : mapping Java/XML

- JAXB : Java Architecture for XML Binding
- JAXB : passer du monde Java au monde XML et inversement
- Principes du mapping Java/XML : manipuler du XML sans jamais voir de XML
- Générer une classe Java à partir d'un schéma XSD et inversement