

# Formation Web Services

## Développement de Web Services avec CXF et JAX-WS

Référence : WS-03

Durée : 3 jour(s)

### Présentation

De par leur nature distribuée, les systèmes d'information sont confrontés à deux problématiques majeures : la réutilisation et l'interopérabilité. Plusieurs technologies ont tenté de répondre à ces objectifs, les Web Services font partie des derniers arrivés. Nous présenterons les concepts et les principes qui ont fait des Web Services une technologie incontournable. Nous nous appuierons sur le langage Java et les frameworks dédiés afin de mettre en œuvre simplement et rapidement une architecture distribuée pérenne.

### Objectifs

- Comprendre les principes d'une architecture orientée services
- Maîtriser les concepts et les enjeux des Web Services
- Savoir développer des Web Services avec via le framework Java CXF
- Pouvoir construire une architecture distribuée WS-\* et REST

**Répartition:** 40% Théorie, 60% Pratique

**Public:** Architecte, Développeur, Chef de projet

**Pré-requis:** Notions XML/Java

### Programme

#### Introduction aux Web Services

- Qu'est-ce qu'un (Web) Service ?
- Les architectures distribuées
- Granularité des services : réutilisation et interopérabilité
- Principes de conception : contrat, couplage lâche...
- Web Services et SOA
- WS-\* et/ou REST
- WS-I et BasicProfile : assurer l'interopérabilité des Web Services

un système distribué

- WS-ReliableMessaging : fiabiliser l'envoi/réception de messages
- WS-BPEL : orchestrer des Web Services
- Web Services et QoS

#### Java et Web Services

- Aperçu de l'offre Java/Web Services
- JAX-WS : Java et WS-\*
- JAX-RS : Java et REST
- Apache CXF : framework Java/Web Services

**REST : un ensemble de ressources**

- REST : Representational State Transfer
- REST : retour aux sources du Web
- URI et opérations de base : GET/POST/PUT/DELETE
- Comparaison avec les Web Services WS-\*
- RESTful ou REST ?
- Réaliser une architecture RESTful avec Java et CXF

#### XML/XSD : Rappels

- XML/XSD : pivot des Web Services
- XML : langage universel
- XSD : décrire un document
- Namespaces et typage

#### SOAP

- SOAP : protocole RPC des Web Services WS-\*
- De XML/RPC à SOAP
- L'enveloppe SOAP
- La jungle des encodages et des communications SOAP
- Couche transport : HTTP mais aussi JMS, SMTP...
- soapUI : un client SOAP
- Développer des Web Services avec CXF
- MTOM : gérer les formats binaires

#### WSDL : le contrat de services

- WSDL : Web Services Description Language
- De l'importance du contrat dans la SOA
- Structure d'un WSDL
- Définition abstraite : quelles opérations sont disponibles ?
- Définition concrète : comment appeler les opérations ?
- WSDL First ou Java First ?

#### UDDI : l'annuaire des services

- UDDI : Universal Description Discovery and Integration
- Où trouver les services ?
- Accéder à l'annuaire
- Administrer un annuaire
- Apache jUDDI : un annuaire simple




#### Web Services asynchrones

- Comment un réaliser un Web Service asynchrone ?
- Callback et/ou polling
- WS-Addressing : s'abstraire de la couche de transport
- Principe de Publish/Suscribe

#### L'univers WS-\*

- WS-Security et problèmes de sécurité des Web Services
- WS-TXM : gérer des transactions courtes/longues dans

## OFFERT EN INTER-ENTREPRISE

-  Le petit déjeuner croissants, jus d'orange, café)
-  Le déjeuner
-  Une qualification téléphonique si nécessaire avec l'un de nos consultants

Tel: +33(0)1 45 26 19 15  
Fax : +33(0)1 75 43 49 92  
Email : training@zenika.com

### Auteur du cours



Grégory Le Bonniec est consultant Java et responsable de l'agence ouest de Zenika. Son parcours professionnel l'a amené à travailler sur de nombreux projets JEE notamment dans le domaine de la SOA. Il est également formateur et conférencier sur différentes technologies comme BPEL, Web Services, Flex...

### Livre offert ! (\*)



Java Web Services:  
Up and Running

(\*) Les livres sont offerts uniquement pour les formations inter-entreprise. Zenika se réserve le droit de changer le livre proposé à tout moment.

