

La norme XML est totalement intégrée par les nouvelles technologies de l'information. Elle permet de fédérer les systèmes d'entreprises en facilitant la communication. Les Web Services apportent quant à eux une réponse efficace aux besoins d'intégration inter-applications tout en supportant le modèle d'architecture SOA centré sur l'accès aux services. En ce domaine, Orsys vous propose des filières complètes pour comprendre les concepts afférents à ces technologies et apprendre les meilleures méthodes pour les mettre en œuvre au sein de vos SI.

Séminaires

[XML, synthèse....](#) (p2)

[Services Web, synthèse....](#) (p4)

[SOA, architecture orientée services, synthèse....](#) (p6)

Stages Pratiques

[Technologies XML, conception, mise en œuvre....](#) (p8)

[XML, modélisation....](#) (p10)

[Développer avec XSL....](#) (p12)

[Web 2.0, développer des applications....](#) (p14)

[Développer des applications XML en Java....](#) (p16)

[Préparer un cahier des charges SOA pour MOA....](#) (p18)

[SOA, conception d'une architecture....](#) (p19)

[SOA, modélisation des processus métiers....](#) (p21)

[Modéliser une architecture SOA avec SOAML....](#) (p22)

[SOA, mise en œuvre en Open Source/Java....](#) (p23)

[Réaliser des processus métiers avec BPMN BPEL/Apache ODE....](#) (p25)

[Jboss Rules, mettre en œuvre des règles métiers....](#) (p26)

[Biztalk 2009, mise en œuvre....](#) (p27)

[Développer des Web services en Java....](#) (p29)

[Web services RESTful, développement avancé en Java....](#) (p31)

[Web services SOAP, développement avancé en Java....](#) (p32)

[.NET 3.5, communications entre applications....](#) (p33)

Séminaire de 2 jour(s)
Réf : XML

Participants

Ce séminaire s'adresse à tous ceux qui souhaitent prendre connaissance du standard XML et découvrir ses nombreuses possibilités d'utilisation dans les systèmes d'information.

Pré-requis

Bonnes connaissances informatiques dans le domaine du Web.

Prix 2012 : 1775€ HT

Eligible DIF

Dates des sessions

Paris

14 juin 2012, 2 oct. 2012
29 nov. 2012

XML, synthèse de la modélisation de contenus aux services Web

OBJECTIFS

XML est devenu un standard incontournable d'échange de données sur le Web. Supporté par toute l'industrie informatique, il a pris une place majeure dans les applications Web, les systèmes d'information, l'intégration de données et d'applications, le commerce électronique B2B... XML fournit toute une galaxie de langages pour la définition et la manipulation de données, mais aussi un cadre pour des architectures distribuées à base de Web services. Ce séminaire offre une vue d'ensemble de XML pour vous permettre d'apprécier son intérêt pour votre système d'information et vos applications Web, et de procéder à l'analyse des différentes solutions.

[1\) Concepts de base](#)

[2\) Définition de données XML](#)

[3\) Manipulation de données XML](#)

[4\) Web Services](#)

[5\) Bases de données XML](#)

[6\) XML et intégration métier](#)

[7\) XML et Web 2.0](#)

1) Concepts de base

Origine et objectifs

- Langages à balises. SGML, HTML et CSS. Forces et faiblesses. Structure, contenu et présentation de documents. L'apport de XML.

Le standard XML

- W3C et xml.org. Les documents composites. Le balisage, les éléments et les attributs. Les espaces de noms xmlns. Les hyperliens avec XLink.
- La galaxie des dialectes XML.

Exemples d'applications

- Gestion documentaire, échange de données et ETL. Publication et diffusion de données.

2) Définition de données XML

DTD et typage des documents

- Notion de document bien formé et valide. Types de base. Déclaration des éléments et des attributs. Insuffisances des DTD.

XML Schema et la modélisation des données

- Types simples et types complexes. Constructeurs de collections. Héritage de types. Réutilisation de déclarations xsd.

Conception des données

- Concevoir, documenter et maintenir DTD et schémas. Règles d'écriture et méthodes de construction de schémas.
- Passage d'UML à XML. Principaux outils.

3) Manipulation de données XML

Publication de données avec XSL

- XPath et l'adressage XML.
- Définition des feuilles de style XSLT. Programmation par règles de production. Les processeurs XSLT.
- Formatage des documents avec XSL/FO.

Traitement de XML

- Traduction de documents XML en graphes d'objets avec l'API DOM.
- Traitement sur flots d'événements avec l'API SAX. Utilisation avec des objets Java.

Architectures de publication

- Interprétation XSL sur le client ou le serveur. Processeurs XSLT et échange de données.
- Gestion de contenus (CMS). Syndication d'information, architecture push, flux RSS ou ATOM.

Les environnements de développement

- Principaux outils de traitement XML (XMLSpy, Oxygen, etc). Avantages des outils de génération XSLT, XSLFO.
- Les ateliers intégrés : Studio Altova, Adobe Studio.

4) Web Services

Architecture des Web services

- Client, fournisseur, annuaire de services. Intégration d'applications.

WSDL, UDDI et SOAP

- Description des services avec WSDL. Référencement des services avec l'annuaire UDDI.
- Le RPC XML. SOAP (Simple Object Access Protocol) : comparaison avec REST.

Composition de services

- Processus métiers et workflows. Transactions longues et composition de services.
- Les standards BPEL, WS-Transaction et WS-Coordination.

Sécurité

- Authentification, XML Encryption, XML signature, XKMS.
- PKI et émergence du standard WS-Security.
- Support dans les serveurs d'application.
- La stratégie .NET. Les serveurs J2EE (Websphere, Weblogic, JBoss, etc.).

5) Bases de données XML

Stockage de XML

- XML comme modèle de données. Mapping relationnel ou format natif. Indexation, recherche par structure ou par contenu. Intégration avec l'existant relationnel.

Langages de requêtes

- Extension de SQL avec SQL/XML. LINQ de Microsoft. Le standard XQuery du W3C. XUpdate pour les mises à jour.

Les produits

- Extension des SGBD relationnels : IBM, Oracle, Microsoft. Les SGBD natifs XML : Tamino, GoXML, TextML, NeoCore, X-Hive, etc. Quid des performances ?
- Etude de cas : PCWorldOnline.

6) XML et intégration métier

Intégration de données via EII XML

- Les EII (Enterprise Information Integrators) basés sur XML : schémas importés et vues intégrées.
- Extraction de guides de données. Décomposition de requêtes. Unification des résultats.
- Exemples d'EII XML : BEA Liquid Data, IBM Information Integrator, Médience (BO).

Intégration d'applications via EAI XML

- Fonctions d'un EAI : connexion, transformation, échange de données, modélisation métier, gestion de workflows. Bus EAI.
- Exemples d'EAI XML : MS BizTalk, BEA Weblogic Integration, IBM Websphere Business Integration, Oracle, Sun ONE, Tibco, etc.
- Intégration d'application via Web services : l'exemple de Qwest avec UDDI.

B2B avec XML

- Apports de XML pour le B2B. Exemples de protocoles : cXML, xCBL, UBL, RosettaNet, OAGIS, FIX.
- ebXML comme remplaçant de l'EDI : les modèles de processus métiers, les activités et les dictionnaires. L'offre ebXML.
- Etude de cas ebXML : le système de e-invoicing de MTR.

7) XML et Web 2.0

Le Web 2.0

- Vers un Web coopératif et social.
- Outils de développement et clients riches.
- Emergence des langages XML pour la programmation (Flex, Silverlight).

Web sémantique

- Description et annotation des pages XHTML et des ressources. Le modèle sémantique RDF.

Participants

Ce cours s'adresse aux responsables de projets et développeurs professionnels désirant comprendre ce que sont les Services Web, les protocoles associés et la manière de les mettre en œuvre en Java et en .NET.

Pré-requis

Connaissances de base en architecture et développement logiciel, et plus particulièrement sur les applications Internet.

Prix 2012 : 1775€ HT

Eligible DIF

Dates des sessions

Paris

24 mai 2012, 7 juin 2012
4 oct. 2012, 20 déc. 2012

Services Web, synthèse

OBJECTIFS

Les Services Web sont un outil d'intégration interapplications et d'ouverture vers l'intranet, l'extranet ou l'Internet, et constituent le moyen permettant de développer des applications distribuées et interopérables. Ce séminaire présente une étude approfondie des Services Web et des protocoles associés : SOAP, WSDL, annuaire UDDI, protocole de sécurité et d'authentification, extensions WS-X. Il vous montre également les principaux outils de réalisation et d'exploitation en environnement Java et .NET.

[1\) Introduction aux Services Web](#)

[2\) Référencement des Services Web](#)

[3\) Fonctionnement des Services Web](#)

[4\) Gestion de la sécurité des Services Web avec WS-Security](#)

[5\) Les extensions WS-X](#)

[6\) Architecture et développement des applications à base de Services Web](#)

[7\) Les Services Web dans l'offre Java/JEE](#)

[8\) Les Services Web dans .NET](#)

1) Introduction aux Services Web

- Technologies de distribution d'applications : de l'approche client/serveur aux objets distribués. Limite dans l'approche objet distribué. Principe des Services Web.
- Architecture des Services Web : protocole d'échange XML, découverte du contrat SOAP, description du Web Service WSDL (Web Service Description Language).

2) Référencement des Services Web

- Annuaire UDDI : principe, fonctionnement, modèle d'information. Utilisation pour le référencement des Services Web. Référencement sur un Intranet et l'Internet.
- Utilisation des annuaires UDDI par les applications : découverte dynamique des services, gestion du catalogue des services.
- Autres types d'annuaires pour les Services Web : utilisation des annuaires de type LDAP, avantages et inconvénients.

3) Fonctionnement des Services Web

- Définition de l'interface des Services Web : utilisation de XML, description de SOAP.
- Méthodes d'invocation des Services Web : appel depuis le client, mise en place d'un proxy Client Web Service.
- Architecture d'applications à base de Web Service : gestion et accès aux données, interaction entre Services Web.

4) Gestion de la sécurité des Services Web avec WS-Security

- Authentification du client : validation de l'identité, transmission de l'identité entre Services Web.
- Gestion de la confidentialité et de l'intégrité des informations circulant entre le client et le serveur.
- Mécanismes d'habilitation et définition des droits du client, WS-Policy.
- Utilisation des en-têtes SOAP pour le transport des informations de sécurité et d'authentification.

5) Les extensions WS-X

- Gestion des transactions avec WS-Coordination.
- Langage d'orchestration : WS-BPEL.
- Suivi de l'acheminement des messages SOAP avec WS-Addressing et WS-ReliableMessaging.
- Gestion de contextes avec WS-Correlation.
- Autres extensions intéressantes : WS-MetadataExchange, WS-CDL, ...

6) Architecture et développement des applications à base de Services Web

- Organisation des applications à base de Services Web.
- Intégration de l'existant : connectivité des Services Web avec les autres technologies. • Intégration des services Web pour des accès via les Services Web.
- Organisation des développements applicatifs utilisant les Services Web. Cycle de test et de validation.
- Applications multi-canaux : accès aux applications depuis le Web, les clients riches et clients légers. Fonctionnement des applications.
- Lien avec les architectures orientées services (SOA).

7) Les Services Web dans l'offre Java/JEE

- Les Services Web en Java selon Javasoft : Core Web Services (JAX-WS, JAXB), Enhanced Web Services (WSIT), Secure Web Services (XWS-Security), Legacy Web Services (JAX-RPC).
- Les projets Web Services Apache.
- Intégration des Services Web dans les outils de développement : Eclipse, Netbeans.
- Plate-forme d'exécution des Services Web : Glassfish & Metro, Bea WebLogic, IBM WebSphere, Apache Axis.

- Exemple de développement avec les Services Web en Java.
- Evolution de l'offre : les acteurs, perspectives et engagement des acteurs sur le marché.

8) Les Services Web dans .NET

- Technologies des Services Web dans .NET. Intégration de SOAP/XML dans le Framework .NET.
- Technologies d'objets à distance : .NET Remoting.
- Plateforme de développement Visual Studio .NET. Mise en place des Services Web. Outils de mise au point. Intégration des clients et des clients Web aux Services Web (applications ASP.NET et Windows Forms).
- Intégration dans l'offre système Microsoft : Serveur.NET, Biztalk, Commerce Server.
- Plateformes d'exécution des Services Web : intégration dans Serveur .NET (Windows 2003 server) des annuaires UDDI, intégration des Services Web.

Séminaire de 2 jour(s)
Réf : SAO

Participants

Ce séminaire s'adresse aux architectes, analystes, concepteurs-développeurs, chefs de projets désirant comprendre les enjeux des architectures de type SOA et les possibilités de mises en oeuvre.

Pré-requis

Connaissances de base en architecture applicative et en technologies Web Services.

Prix 2012 : 1775€ HT

Eligible DIF

Dates des sessions

Paris

31 mai 2012, 4 oct. 2012
6 déc. 2012

SOA, architecture orientée services, synthèse

OBJECTIFS

Ce séminaire présente les enjeux fonctionnels et techniques liés à la mise en oeuvre d'Architectures Orientées Services ou SOA. Il présente également un panorama des meilleures pratiques pour la définition d'une architecture SOA ainsi que le lien avec les Web Services. L'enjeu : passer d'une informatique essentiellement composée d'applications à une informatique orientée services afin d'améliorer la réactivité du système d'information.

1) Introduction

2) Présentation d'une architecture SOA

3) Services et processus

4) Aspects techniques

5) Démarche méthodologique de conception SOA

6) Les acteurs et produits du marché

1) Introduction

- Les enjeux pour l'entreprise.
- Les architectures multiniveaux et les composants métiers : limites de l'interopérabilité des middlewares classiques.
- Services Web et interopérabilité des SI.
- Des composants aux architectures orientées services (SOA) : limites de la gestion par projet du SI, introduction aux services, contrats de service, orchestration de service, bus de services d'entreprise (ESB), etc.
- Gains de l'approche SOA.

2) Présentation d'une architecture SOA

- Composants et couches d'un SOA.
- Service d'accès aux données, gestion des transactions, gestion des processus, interaction utilisateur (portail...), sécurité, administration, supervision, etc.

3) Services et processus

- Le concept de service.
- Exposition de service, couplage faible, synchronisme vs asynchronisme, fournisseur et consommateur de services, contrat de service, typologie des services (métier, technique, etc.), différences entre services et composants, spécification de la qualité de service.
- Composant applicatifs métiers.
- Unité d'exploitation dans les SOA, implémentation des contrats, dépendances entre composants et orchestration.
- Mise en correspondance processus métiers/services.

4) Aspects techniques

- Mise en oeuvre de Web Services (JEE, .NET, PHP, etc.).
- Les fondations XML.
- Schéma XML pour l'interopérabilité et la description des données applicatives.
- Description des services avec WSDL.
- Invocation avec SOAP.
- Rechercher et publier des services (annuaires).
- Design patterns liés aux Services Web.
- Lien entre SOA et EAI.
- Middlewares orientés messages (MOM).
- Enterprise Service Bus : notion d'ESB, ESB vs SOA. Le standard JBI (Java Business Integration).
- Orchestration de services web et intégration de processus métiers (BPM, BPEL, ...), gestion d'applications composites (SCA : Service Component Architecture).
- Les standards et leur niveau d'implémentation : Sécurité et Services Web (WS-Security), gestion des transactions (WS-Transactions), délivrance des messages (WS-ReliableMessaging), interopérabilité et identification des messages (WS-Addressing).

5) Démarche méthodologique de conception SOA

- Les liens entre SOA et l'approche objet.
- Les différentes approches et méta modèles (RUP, PRAXEME, etc.).
- Cycle de vie d'un projet SOA : vision stratégique et processus organisationnel.
- Métaphore de l'urbanisation et niveaux d'agrégation.
- Modèle conceptuel d'un SOA.
- La modélisation des services au sein de l'architecture applicative avec UML.
- Le passage du processus organisationnel aux services métiers, des services métiers aux services applicatifs.
- L'approche MDA de l'OMG.

6) Les acteurs et produits du marché

- Typologie des produits existants.
- Les Enterprise Service Bus (ESB).
- Les plates-formes EAI pour les nouvelles fonctionnalités Web Services (SeeBeyond, Tibco, WebMethods, etc.).
- Les orchestrateurs SOA, les couches hautes pour les annuaires, l'orchestration, l'administration.
- Les fournisseurs de plates-formes (IBM, BEA, Oracle, etc.).
- L'offre et les projets Open Source : JEMS, Mule, Cetix, Synapse...

Stage pratique de 4 jour(s)

Réf : PXM

Pré-requis

Connaissances de base en développement, bases de données, architectures Internet/Web.

Prix 2012 : 2150€ HT

Eligible DIF

Dates des sessions

Paris

22 mai 2012, 19 juin 2012
10 juil. 2012, 28 août. 2012
18 sep. 2012, 16 oct. 2012
13 nov. 2012, 11 déc. 2012

Aix

25 sep. 2012, 20 nov. 2012

Bordeaux

11 juin 2012, 9 oct. 2012
11 déc. 2012

Bruxelles

9 oct. 2012, 11 déc. 2012

Genève

9 oct. 2012, 11 déc. 2012

Lille

9 oct. 2012, 11 déc. 2012

Luxembourg

9 oct. 2012, 11 déc. 2012

Lyon

25 sep. 2012, 20 nov. 2012

Nantes

11 sep. 2012, 20 nov. 2012

Rennes

11 sep. 2012, 20 nov. 2012

Sophia-antipolis

25 sep. 2012, 20 nov. 2012

Strasbourg

9 oct. 2012, 11 déc. 2012

Toulouse

11 juin 2012, 9 oct. 2012
11 déc. 2012

Technologies XML

conception et mise en oeuvre

OBJECTIFS

XML est devenu incontournable tant sur le poste utilisateur que sur les serveurs de l'entreprise. Le besoin de partage et d'échange d'informations entre utilisateurs et entre applications a accéléré l'usage des technologies XML. Ce stage vous montrera comment tirer profit de ces technologies et des normes associées. Au moyen de travaux pratiques, vous apprendrez à concevoir et à gérer vos données XML, à les diffuser et à les publier, à les exploiter via vos applications, vos services Web et vos bases de données, sans négliger les aspects liés à la sécurité.

1) Industrie XML

2) Les parseurs XML

3) Validation de documents

4) Transformation XSLT

5) Publication de documents

6) Message et échanges XML

7) XML et les bases de données

8) Sécurité des échanges XML

9) Interfaces graphiques

1) Industrie XML

- Rappel sur les langages structurés : SGML, HTML, XML et XHTML.
- Le contexte actuel. Pourquoi XML est incontournable.
- XML et le schéma directeur du S.I. La structure XML.
- Avantages et inconvénients de XML.
- Rôle prédominant des offres bureautiques.
- Les langages XML techniques (WML, SVG, VoiceXml) et les langages XML métier (FixXML, CML...).
- Panorama des éditeurs XML.

2) Les parseurs XML

- Offre du marché. Rôle des parseurs : performances, avantages et inconvénients.
- Les types de parseurs : push (SAX), pull (XMLPull ou StAX). Les représentations arborescentes (DOM, JDOM, ...), "Data Binding" (Castor ou JAXB).
- Les Data Island et l'intégration dans les pages Web.
- Les accélérateurs XAN (XML Aware Network).
- Les parseurs Java (JAXP), .Net, Flash (XML Parser).
- Ajax et la lecture de données dans les pages HTML.

Travaux pratiques

Mise en oeuvre de différents parseurs.

3) Validation de documents

- Structure et composants d'un document XML : éléments, attributs, entités...
- Les méta-données (XMP, IPTC, EXIF).
- Document bien formé et document valide.
- Les modèles de validation. DTDs, XML-schema (XSD), Relax-NG, Schematron. Syntaxe et mise en oeuvre. Exemple : DTD de l'édition (MARC, TEI).
- Schémas XSD : structure, syntaxe, typage des données, outils de composition. Modélisation.
- Les Namespaces. Rôle, intégration, partage, création.
- Exemple d'appel à partir d'une page XHTML.

Travaux pratiques

Création de document XML, validation de document. Création de documents DTD et XSD. Association de Namespaces aux DTD et XSD.

4) Transformation XSLT

- Enjeux du langage XSLT.
- Intégration dans le système d'information.
- Le langage de transformation XSL-T 1.0 et 2.0 : construction d'arbres, restructuration, génération multiformats : XHTML, SVG, PDF.

5) Publication de documents

- Visualisation brute sous les navigateurs.
- Mise en forme avec les feuilles de style CSS.
- Rôle du XHTML sur l'accessibilité. Conseils techniques. Outils de validation du W3C.
- XPATH 1.0 et 2.0 : naviguer dans les données XML.
- Les feuilles de style XSL-T : templates, structures. Mode de programmation déclaratif, impératif, hybride.
- Le langage XSL-FO pour le formatage d'objets.
- Conversion en format papier PDF, RTF... Avantages et usage des Outils de générations de XLS (Stylevision)

Travaux pratiques

Création de feuille de style en XSL-T et XSL-FO.

6) Message et échanges XML

- Le DOM du W3C et l'interface de programmation normalisée : Data Island, Ajax, JavaScript.
- Les serveurs d'applications : .NET et JEE.
- Portail d'entreprise.
- Flash : FLEX (MXML), échanges XML et services Web.
- Transfert et sérialisation de messages : Rest, XML-RPC, SOAP, WSDL, UDDI. Définition des Services Web.
- XML et la bureautique. Les formats d'échange de documents. RDF. Offre Office de Microsoft. Open XML, OOXML. Office Open XML vs OpenDocument.

Travaux pratiques

Appel Ajax et traitement JavaScript de documents XML. Interrogation d'un Service Web.

7) XML et les bases de données

- Positionnement du XML par rapport aux bases de données. Stockage de documents XML. Bases semi-structurées. Data Island.
- SGBDR aux bases natives XML Xindice, Tamino.
- Outils de Mapping pour la production de codes C#, java, XSL dédiés aux conversions de types (SOAP / SGBD / XML / XSD)
- Langages d'interrogation XML : XPath, XQuery, extensions du SQL.
- XML et les bases (Oracle , SQL Server 2005, MySQL5).

8) Sécurité des échanges XML

- Les différents protocoles de sécurisation. HTTPS, XML Signature (Xml Dsig), XML encryption, XML access control, WS-security.

Travaux pratiques

Exemples des différentes technologies présentées.

9) Interfaces graphiques

- Le langage XAML et Silverlight de Microsoft.
- Le langage XUL de l'Open Source.
- SMIL : les animations Web.
- Le langage MXML de Flex d'Adobe
- XForms, InfoPath : la technologie de formulaires électroniques.
- Blogs et RSS (Really Simple Syndication), ATOM un format de syndication de contenu Web.
- SVG : standard vectoriel.

Travaux pratiques

Mise en oeuvre des technologies XUL, XForms, RSS.

Stage pratique de 3 jour(s)
Réf : CAX

Participants

Architectes, concepteurs d'applications, développeurs, chefs de projets et maîtres d'ouvrage informatique. Tous ceux qui ont à concevoir des modèles XML pour les données et les documents de leurs systèmes d'information.

Pré-requis

Connaissances de base des technologies XML et de la modélisation de données.

Prix 2012 : 1730€ HT

Éligible DIF

Dates des sessions

Paris

4 juin 2012, 1 oct. 2012
3 déc. 2012

XML, modélisation concevoir et structurer vos documents et données

OBJECTIFS

Ces dernières années ont vu une forte progression de la popularité de XML, le standard du W3C pour l'écriture de documents structurés. Les différentes options proposées par les recommandations du W3C seront ici analysées en détail et illustrées par des exercices et des études de cas qui permettront d'apprendre à modéliser correctement des documents XML. A l'issue de cette formation, le participant aura une maîtrise complète de la syntaxe du langage XML Schéma et des règles de modélisation permises par ce langage.

1) Rappels sur XML et les technologies associées

2) XML et la composition de documents

3) Les espaces de noms

4) Le langage XML Schema

5) Techniques de modélisation XML

6) XML et la conception objet

7) Les meilleures pratiques de modélisation

1) Rappels sur XML et les technologies associées

- Structure d'un document XML, arborescence des éléments.
- Composants d'un document XML : déclarations, instructions processeur, commentaires, entités, éléments, attributs.
- Document bien formé et document valide.
- Règles pour les documents bien formés, structure logique d'un document XML.
- Les données structurées et semi-structurées.
- Les différents types de parseurs XML et leur rôle.
- La galaxie des technologies XML.

Travaux pratiques

Création et validation d'un premier document XML.

2) XML et la composition de documents

- Objectifs des langages de structuration : DTD, XML Schema, Schematron et Relax NG.
- Pourquoi un schéma est-il nécessaire.
- Comment utiliser des schémas.
- Principales caractéristiques des schémas.
- Les principales parties d'un schéma.
- Comment fonctionnent les schémas.
- Les différents langages de modélisation.
- Syntaxe EBNF pour DTD (Data Type Definition).
- Validation des documents.
- Les différents types d'entités et leur utilisation.
- DTD à racines multiples.
- Extensibilité et réutilisation de composants du modèle.
- Les limites du langage.

Travaux pratiques

Écriture d'une DTD et de documents conformes à celle-ci. Restructuration modulaire de la DTD.

3) Les espaces de noms

- Description des espaces de noms.
- Utilisation des espaces de noms.
- Les Namespaces et la combinaison de vocabulaires multiples.
- Combinaison de langages : XHTML, Xforms et SVG.
- Les erreurs à ne pas commettre.
- Étude de la portée d'un espace de nom.
- Définition d'une URI.

Travaux pratiques

Écriture d'un document utilisant plusieurs espaces de noms.

4) Le langage XML Schema

- La syntaxe du langage.
- Définition d'un vocabulaire (éléments, attributs, etc.) et d'une grammaire.
- Validation des documents.
- Tester les valeurs des éléments et des attributs.
- Étude des types de données simples et complexes.
- Type de base du langage et création de type personnalisé.
- Extension et restriction des types.
- Organisation des éléments : séquence, choix, tas.

- Les éléments et les groupes abstraits de substitution.
- Utilisation des clés et des règles d'unicité.
- Définition d'espaces de noms.
- Schémas stricts et schémas permissifs.

Travaux pratiques

Redéfinition d'une DTD avec la syntaxe XML-Schema. Amélioration du code et définition de types de données.

5) Techniques de modélisation XML

- Les modèles de conception.
- Schémas uniques applicables à différents types de documents.
- Modèles génériques réutilisables (liste, grille, etc.).
- Documents utilisant de multiples schémas.
- Création de schémas extensibles et contrôle de l'extensibilité.

Travaux pratiques

Application des modèles étudiés sur différents cas.

6) XML et la conception objet

- Rappels sur les concepts de l'orienté objet.
- Concepts objets du langage XML Schema.
- Types, classe, dérivation, polymorphisme.
- Représentation UML/XML.
- Inclusion de schémas.
- Importation de schémas.
- Redéfinition de schémas.

Travaux pratiques

Composition de schémas modulaires et réutilisables.

7) Les meilleures pratiques de modélisation

- Composition et héritage.
- Types de constructions : "Poupées russes", "Tranches de salami" et "Stores vénitiens".
- Schémas d'adaptation.
- Définition de conteneurs et d'éléments.
- Catalogues de modèles de conception.
- Localisation ou exposition des espaces de noms.
- Documents homogènes et hétérogènes.
- Définition globale ou locale des composants.
- Travailler avec des éléments ou des types.
- Choix du Namespace par défaut.
- Définition de conteneurs à contenu variable.
- Des modèles de contenu extensible.
- Etendre XML Schema avec XSLT et Schematron.
- Alternative Relax NG.

Travaux pratiques

Evaluation des différentes méthodes de composition et utilisation appropriée des différents langages.

Stage pratique de 4 jour(s)

Réf : LSX

Pré-requis

Connaissances de base en XML et de la syntaxe HTML.

Prix 2012 : 2150€ HT

Eligible DIF

Dates des sessions

Paris

29 mai 2012, 18 sep. 2012

20 nov. 2012

Développer avec XSL

génération de documents XML, HTML, graphiques et pdf

OBJECTIFS

Cette formation vous permettra de maîtriser les langages XSL-T et XPath et de réaliser des feuilles de style en utilisant des techniques avancées. Elle vous montrera comment utiliser le langage XSL-FO pour la génération de documents PDF ainsi que le langage SVG pour la génération d'images et d'interfaces sophistiquées. Les participants mettront en place un environnement complet de production automatisée de documents contenant à la fois textes et images.

1) Introduction au contexte de XSL et XPath

2) Le langage d'interrogation : XPath

3) Le langage de transformation : XSL-T

4) XSL-T, fonctionnalités avancées

5) La boîte à outils du développeur

6) XSL-T 2.0, X-PATH 2.0

7) SVG pour les interfaces graphiques et la création d'images

8) XSL-FO pour la génération de documents papier

1) Introduction au contexte de XSL et XPath

- Besoins par rapport à l'utilisation de la technologie XML. Technologie CSS pour la présentation, limites de CSS. Introduction à XSL, XSL-T, XSL-FO et SVG. Introduction à Xpath. Principes et mécanismes de base. Parseur XML et processeur XSL. Support par les navigateurs Web. Mise en œuvre des technologies XSL.

2) Le langage d'interrogation : XPath

- Expressions XPath. Interrogation d'un document XML.
- Axes et vecteurs de recherche.
- Revue de quelques expressions simples.
- Sélection d'éléments/nœuds, de jeux de nœuds et d'attributs dans un arbre XML.
- Opérateur de localisation.
- Equation d'égalité et de relation, opérateurs booléens. Opérations numériques.
- Fonctions standard et extensions des processeurs.

Exercice

Ecriture et tests d'expressions XPath.

3) Le langage de transformation : XSL-T

- Structure d'une feuille de style XSL-T, traitement du processeur XSL-T.
- Namespaces et éléments XSL racine.
- Formats de sortie : vocabulaires XML, HTML, X-HTML ou texte.
- Utilisation d'expressions XPath.
- Nœuds et jeux de nœuds.
- Modèle de traitement : les templates.
- La syntaxe du langage XSL-T, les types de données, les structures de contrôle, etc.
- Notions de variables et de paramètres.
- Génération d'éléments et d'attributs.
- Invocation déclarative ou impérative des traitements.
- Récupération d'éléments et de contenu.
- Analyse du comportement d'un processeur XSL-T.
- Optimisation des recherches de nœuds.
- Traitement des erreurs.

Exercice

Composition de différents types de feuilles de style XSL-T.

4) XSL-T, fonctionnalités avancées

- Ecriture de fonctions : appels simples et récursifs.
- Structure modulaire des feuilles de style.
- Les design patterns : Fill-in-the-blank, Navigational, Rule-based, Computational.
- Import et inclusion de feuilles de style avec passage d'arguments.
- Création de bibliothèques réutilisables.
- Modularité et héritage.
- Transformations basées sur des sources multiples.
- Générations de groupes de documents.
- Spécificités de certains processeurs.
- Travail sur des espaces de noms hétérogènes.

Exercice

Création de feuilles de style modulaires et imbriquées, avec de multiples structures en entrée et génération de plusieurs documents en sortie.

5) La boîte à outils du développeur

- Lancement de transformations en ligne de commande.
- Utilisation de divers processeurs (XT, Saxon, Xerces, MSXML...).
- Ecriture de script de transformation avec Ant.
- Gestion de site Web.

Exercice

Mise en place d'un environnement de gestion de site Web basé sur XML.

6) XSL-T 2.0, X-PATH 2.0

- Nouveautés : fonctions, expressions, opérations, tris, fallbacks et traitements des erreurs, etc.
- Nouveaux types de données supportés.
- Différences de traitement.
- Création d'arbre résultat, d'une séquence de nœuds ou de valeurs.
- Requêtes XPath avec expressions régulières.
- Les fonctions et opérateurs sur les numériques.
- Les fonctions sur les chaînes de caractères.
- Les fonctions et les opérations sur les dates.

7) SVG pour les interfaces graphiques et la création d'images

- SVG, Scalable Vector Graphics. Les enjeux des langages vectoriels. Avantages de SVG sur Flash. Rôle et importance de SVG.
- Utilisation côté serveur et côté client.
- Environnements d'édition et d'utilisation : navigateurs... Les systèmes de coordonnées et le dimensionnement. Les formes de base. Structure de document. Les styles. Transformation du système de coordonnées. Les courbes et les chemins. Motifs et dégradés. Texte et internationalisation. Masquage et découpe.
- Filtrage.
- Animation, interaction et scripts.

Exercice

Création de feuilles de style permettant de générer des documents SVG. Génération de pages HTML contenant des graphes.

8) XSL-FO pour la génération de documents papier

- XSL-FO, Formatting Objects.
- Besoins et objectifs de XSL-FO. Introduction au langage.
- Solutions et implémentations disponibles.
- Exemple à partir du projet FOP de Apache.org.
- Eléments du langage XSL-FO : initialisation, définition de modèles de pages, la zone de contenu, dimensions des régions, séquences des pages, génération du résultat.
- Inclusion de graphiques au format SVG.

Exercice

Création de feuilles de style combinant XSL-T et XSL-FO pour générer des sorties en PDF. Génération d'un rapport contenant des images au format SVG.

Stage pratique de 4 jour(s)
Réf : XAW

Participants

Ce cours s'adresse aux webmasters, développeurs web, informaticiens.

Pré-requis

Bonnes connaissances du Web, d'HTML et de CSS.
Connaissances de base du JavaScript et du XML.

Prix 2012 : 2150€ HT

Eligible DIF

Dates des sessions

Paris

12 juin 2012, 11 sep. 2012
13 nov. 2012

Web 2.0, développer des applications

OBJECTIFS

Web 2.0 est devenu un standard pour la définition des applications Web de demain et l'architecture des services associés. Ce cours vous apprendra, au moyen de nombreux travaux pratiques, à maîtriser les règles de conception et les technologies nécessaires à la réussite de votre site Web 2.0, HTML5.

[1\) Rappels XML](#)

[2\) Introduction au Web 2.0](#)

[3\) Feuilles de style](#)

[4\) Interfaces graphiques](#)

[5\) Interaction et formulaires](#)

[6\) JavaScript](#)

[7\) Les Frameworks professionnels](#)

[8\) Les solutions Web 2.0](#)

1) Rappels XML

- Structure XML et documents valides.
- Savoir lire un schéma XSD.
- Web sémantique et référencement (SEO, SiteMap).

Travaux pratiques

Création de documents XML, de XSD et de XSL. Tests d'accessibilité. Création de page XHTML.

2) Introduction au Web 2.0

- Les concepts et les technologies Web 2.0.
- Règles de conception.
- Navigateurs du marché, impact sur la portabilité.

Travaux pratiques

Analyse d'ergonomie Web 2.0. Interrogation de Services Web à partir du Web.

3) Feuilles de style

- Les feuilles de style CSS, CSS-3 : couplage avec XML.
- Le développement d'interface riche et animée.

Travaux pratiques

Paramétrage automatique de blocs, calques, menus, zones de formulaires à partir de CSS.

4) Interfaces graphiques

- Langages XUL, XAML/SilverLight, Flex/MXML, SVG, HTML5.
- Flash pour traiter des données XML distantes.
- Présentation des technologies RIA et RDA.

Travaux pratiques

Développement d'une application de base XUL, Flex, Flash. Conversion HTML vers PDF. Création de gadgets.

5) Interaction et formulaires

- Formulaires XForms dans les interfaces Web 2.0.
- Offre Microsoft InfoPath.
- les solutions de contrôle de saisie (autosuggest,...)

Travaux pratiques

Démonstration et création de formulaires.

6) JavaScript

- Variable, fonctions. Manipulation d'objet.
- manipulation du DOM pour gérer l'interactivité.
- Portabilité, industrialisation des programmes.
- Générateur d'interface GWT.
- Environnements de développement et de test.
- Ajax. Principe. Codage.

7) Les Frameworks professionnels

- Google API (Google Maps, Ajax, feed, ...).
- Prototype, jQuery, Mootools, Yahoo (YUI), Adobe (SPRY).

Travaux pratiques

Création d'une cartographie Google Maps.

8) Les solutions Web 2.0

- Définitions (blog, audioblog, podcast, outils...).
- WordPress pour la création et gestion de ses blogs.
- Les protocoles RSS, ATOM dans les blogs.
- Les Wikis. Les solutions de commerce 2.0.

Travaux pratiques

Création d'un blog et de son flux RSS.

Stage pratique de 4 jour(s)

Réf : XDJ

Pré-requis

Bonnes connaissances du langage Java et des technologies Web. Expérience requise en développement Java.

Prix 2012 : 2150€ HT

Eligible DIF

Dates des sessions

Paris

4 sep. 2012, 20 nov. 2012

Développer des applications XML en Java

OBJECTIFS

Java et XML représentent une très bonne combinaison pour le développement de projets de systèmes d'information distribués. Cette formation, destinée aux développeurs Java, leur apportera toutes les connaissances nécessaires à la mise en oeuvre des technologies associées à XML dans l'environnement Java

[1\) Introduction à XML et aux technologies associées](#)

[2\) XML et la composition de documents](#)

[3\) Exploitation de documents XML en Java](#)

[4\) Java et XML dans la communication entre composants](#)

[5\) Fonctions avancées](#)

[6\) Base de données](#)

1) Introduction à XML et aux technologies associées

- Rappel sur les langages à balises : SGML, HTML, XML et XHTML.
- XML et le concept de documents.
- Domaines d'application de la technologie XML.
- Technologies associées : Namespace, les schémas DTD et XML-Schema, CSS, XSL-T et XPath, XLink et XPointer, XSL-FO.
- Les parseurs/processeurs XML et XSL.
- Java et XML : XML au sein de la plate-forme JEE.

2) XML et la composition de documents

- Structure d'un document XML, arborescence des éléments.
- Pièces d'un document XML : déclarations, instructions processeur, commentaires, entités, références caractères, éléments, attributs.
- Document bien formé et document valide.
- Règles pour les documents bien formés, structure logique d'un document XML.
- Les schémas à l'aide de DTD et XML-Schema.
- Les namespaces.
- Utilisation de feuille de style CSS pour présenter les documents.
- Le langage XSL-T pour la transformation des documents.

Exercice

Composition et exploitation de documents XML bien formés et valides, avec les schémas associés. Les documents seront réalisés à l'aide d'un éditeur simple et d'outils spécialisés.

3) Exploitation de documents XML en Java

- Introduction et historique des parseurs XML pour Java.
- Exploitation des différents parseurs en Java.
- Interface de programmation DOM, SAX et JDOM.
- Situation du marché, différents parseurs disponibles, l'API JAXP.
- Choix du parseur approprié.
- Consommation et validation de document XML.
- Production de documents XML bien formés et valides.

Exercice

Développement d'applications de traitement et de production de contenu XML.

4) Java et XML dans la communication entre composants

- Principe de base des applications distribuées.
- Rôle de XML dans l'échange d'information.
- Web Services.
- Exemple d'une application distribuée utilisant SOAP et WSDL.
- Echanges de flux XML.

Exercice

Développement d'un Web service sur serveur Java et appel depuis un client Web.

5) Fonctions avancées

- Mapping d'objets Java et de documents XML (JaxB, Castor...).
- Utilisation de XSL-T pour la transformation de documents XML à partir de Java.
- Programmation par transformations de flots XML.
- Association Objet et XML.
- Utilisation de XSL-FO pour la publication de document PDF.
- Compilation de feuilles de style (XSLTC).
- Génération d'images depuis des documents XML (SVG - Batik).
- Optimisation des performances.

Exercice

Génération de graphiques par feuille de style compilée et mesures de performances.

6) Base de données

- Architecture générique d'un système XML/Relationnel.
- Modèle de données et stockage du DOM.
- Reconstruction d'un document depuis la base.
- Injection de documents.
- Interprétation d'expressions Xpath.

Exercice

Manipulation optimisée à travers une base de documents de gros volume.

Stage pratique de 2 jour(s)
Réf : SOM

Participants

Ce cours s'adresse aux analystes et chefs de projets MOA et AMOA impliqués dans la réalisation d'un projet de type SOA.

Pré-requis

Bonnes connaissances en matière de modélisation et d'organisation d'un développement informatique.

Prix 2012 : 1370€ HT

Eligible DIF

Dates des sessions

Paris

14 juin 2012, 4 oct. 2012
13 déc. 2012

Préparer un cahier des charges SOA pour MOA

OBJECTIFS

Ce stage vous apprendra à concevoir et réaliser un cahier des charges dédié à la mise en place d'une architecture SOA. A l'issue, vous maîtriserez les différentes étapes de la démarche, et aurez appris à recenser et représenter les informations, formaliser les concepts et modéliser les processus métiers au moyen de standards comme UML et BPMN.

1) Les Architectures Orientées Services

2) La démarche générale

3) La modélisation des processus métier avec BPMN

4) Les apports d'UML

5) La prise en compte des contraintes techniques SOA

6) Le cahier des charges

1) Les Architectures Orientées Services

- Spécificités des architectures SOA.
- Les différents constituants d'une SOA (Web Services, ESB, XML, BPEL, ...).
- La notion de service et le concept d'orchestration de services.
- La vision induite par la mise en place d'une SOA et l'alignement du SI sur les processus métier de l'organisation.
- L'impact sur le fonctionnement de la MOA et les relations MOA / MOE.

2) La démarche générale

- L'étude de faisabilité : le repérage du domaine, la découverte des informations.
- L'étude de faisabilité : la modélisation des processus métier de l'organisation.
- Les standards communs entre MOA et MOE (UML, BPMN, ...).
- L'apport de ces standards pour formaliser les éléments découverts lors de l'étude de faisabilité.
- Le diagnostic et la reconfiguration du SI.
- La modélisation du SI cible et la rédaction du cahier des charges.

3) La modélisation des processus métier avec BPMN

- La justification de cette étape et l'apport de BPMN en tant que formalisme commun MOA / MOE.
- Les origines de BPMN et l'état de la normalisation.
- La description d'un processus avec BPMN, avec ses acteurs et ses informations associées.
- Les concepts BPMN d'activités, d'événements, de connecteurs, de branchements, ...
- Le passage de BPMN utilisé par la MOA à BPEL utilisé par la MOE.

Travaux pratiques

Modélisation de processus métier en BPMN.

4) Les apports d'UML

- Le diagramme de communications en appui du repérage du domaine.
- Le diagramme de classes pour représenter les concepts d'information.
- Les cas d'utilisation et les diagrammes d'activités en complément de la modélisation des processus métier.
- Les diagrammes de composants et de déploiement pour représenter l'existant de l'architecture du SI.
- Les diagrammes de composants et de déploiement pour modéliser les grandes lignes de l'architecture cible.

Travaux pratiques

Réalisation de diagrammes UML à l'aide d'un atelier.

5) La prise en compte des contraintes techniques SOA

- L'impact de l'état de l'art : standards, extensions, offres éditeurs, ...
- La nécessité de s'appuyer sur une AMOA.
- L'importance des pré-études pour délimiter les caractéristiques techniques.

6) Le cahier des charges

- La description du futur SI : objectifs, cartographie de l'existant et de la cible.
- La description du futur SI : description des entités et processus métiers.
- La définition de la stratégie attendue et des exigences associées.
- La prise en compte des contraintes techniques spécifiques à SOA (CCTP).
- L'usage que la MOE fait des éléments fournis par la MOA.
- Les recommandations en matière de cycle de développement côté MOE.

Travaux pratiques

Rédaction des points importants d'un cahier des charges.

Stage pratique de 4 jour(s)
Réf : SOB

Participants

Chefs de projets, architectes désirant comprendre les enjeux liés aux architectures orientées services et la démarche de mise en œuvre.

Pré-requis

Bonnes connaissances des architectures multiniveaux, de la conception d'applications et d'UML.

Prix 2012 : 2150€ HT

Eligible DIF

Dates des sessions

Paris

22 mai 2012, 19 juin 2012
4 sep. 2012, 20 nov. 2012

SOA, conception d'une architecture

OBJECTIFS

Cette formation vous présentera la démarche méthodologique PRAXEME de mise en œuvre d'une architecture orientée services. Elle vous permettra aussi de comprendre l'architecture des services Web, la problématique d'orchestration de services et le lien avec les approches orientées composants. Les produits des principaux éditeurs et l'offre open source autour des SOA vous seront présentés.

1) Introduction

2) Les concepts liés aux SOA

3) Introduction à l'approche méthodologique

4) Le modèle sémantique du domaine

5) Processus métier et cas d'utilisation du SI

6) Définition de l'architecture logique

7) Définition de l'architecture technique

8) Les acteurs et produits du marché

Travaux pratiques

Ce cours s'appuiera sur une étude de cas permettant d'appréhender par la pratique les différentes phases de la démarche de conception d'un SOA. L'étude de cas sera réalisée avec un AGL.

1) Introduction

- Les enjeux pour l'entreprise et l'approche MDA de l'OMG.
- Architecture multiniveaux, composants métiers : limites de l'interopérabilité des middlewares classiques.
- Services web et interopérabilité des SI.
- Des composants aux architectures orientées services (SOA) : limites de la gestion par projet du SI, introduction aux services, contrats de service, orchestration de service, gains de l'approche SOA.

2) Les concepts liés aux SOA

- Modèle conceptuel d'un SOA.
- Le concept de service métier : exposition de service, couplage faible, synchronisme vs asynchronisme, fournisseur et consommateur de services, contrat de service, typologie des services, différences entre services et composants.
- Composant applicatifs métiers : unité d'exploitation, implémentation des contrats, dépendances entre composants et orchestration.

3) Introduction à l'approche méthodologique

- Les différentes approches et méta-modèles : Framework de Zachman, axes méthodologiques du RUP, l'approche PRAXEME, processus de développement.
- Cycle de vie d'un projet SOA : vision stratégique et processus macro-organisationnel, processus organisationnel, implémentation technique, métaphore de l'urbanisation et niveaux d'agrégation, le vocabulaire PRAXEME : fabrique logique, atelier et machine logique.

4) Le modèle sémantique du domaine

- Analyse du domaine métier.
- Modéliser les objets de références et leur cycle de vie, structuration du modèle et lien avec l'urbanisme.

5) Processus métier et cas d'utilisation du SI

- Introduction au modèle pragmatique.
- Analyse des processus métier : modélisation à l'aide de diagramme d'activité, liaison avec le modèle sémantique, classes pragmatiques.
- Les cas d'utilisation du SI.
- Le modèle géographique.

6) Définition de l'architecture logique

- Dérivations des modèles sémantique et pragmatique vers l'architecture logique : gestion des navigations, cas des relations 1..n, n..m, classe associative, règles de découpage du diagramme de classes.
- Spécification des machines logiques et des services logiques : emploi de la démarche MDA, regroupement en atelier et fabrique logique, lien avec l'urbanisation.
- Définition des services logiques : précondition, post-condition, exception ou signaux, message d'entrée et de réponse, définition des types complexes associés, gestion de variantes et contextualisation.
- Contrat d'utilisation, spécification de la qualité de service, métrique de Qualité.
- Intégration de l'existant.

7) Définition de l'architecture technique

- Implémentation du modèle logique sous la forme de composants : composants et couches d'un SOA, composants processus, service, façade, factory et types de factories, l'approche MDA.
- Gestion des versions, mode d'interaction et message d'entrée/sortie, gestion des transactions et service de compensation.

- Description des services avec WSDL, invocation avec SOAP : construire des schémas XML interopérables, design patterns liés aux services Web.
- L'infrastructure des services Web (WS-*), gestion de la sécurité, des transactions, de la délivrance des messages, utilisation des annuaires UDDI.
- Orchestration de services Web et intégration de processus métiers : présentation des standards BPEL, BPMN : création de représentation visuelle d'un processus et exécution de script BPEL, gestion transactionnelle et contexte d'exécution.
- Gestion d'applications composites : introduction au standard Service Component Architecture (SCA).
- Les Enterprise Service Bus (ESB) : notion d'ESB, introduction au JBI.

8) Les acteurs et produits du marché

- Typologie des produits existants et critères de sélection.
- Présentation de l'offre des principaux acteurs.
- L'offre Open Source.

Stage pratique de 2 jour(s)
Réf : BPM

Participants

Chef de projets MOA/MOE,
analyste et expert métier,
assistant à la maîtrise
d'ouvrage, urbaniste.

Pré-requis

Bonnes connaissances des
composantes d'un système
d'information.

Prix 2012 : 1285€ HT

Eligible DIF

Dates des sessions

Paris

31 mai 2012, 13 sep. 2012
15 nov. 2012

SOA, modélisation des processus métiers BPNM - BPEL

OBJECTIFS

La mise en place d'applications gérant les processus métiers est un enjeu majeur. Avec les architectures orientées services (SOA), de nouveaux outils et méthodes ont émergé pour répondre à ces préoccupations. Ce cours vous les présente et aborde par la pratique les standards liés à la gestion des processus métiers.

1) Introduction

2) Les concepts liés aux processus métiers et aux SOA

3) Introduction à l'approche méthodologique

4) BPMN : Business Process Modeling Notation

5) Le modèle des objets métiers

6) Implémenter un processus avec BPEL

7) Panorama des outils

Travaux pratiques

Ce cours s'appuiera sur une étude de cas permettant d'appréhender par la pratique les différents concepts et étapes présentés. Utilisation d'objecteering et d'outil BPEL.

1) Introduction

- Les enjeux pour l'entreprise de la modélisation des processus métiers.
- Emergence des architectures orientées services, risques et gains potentiels liés à cette approche.
- Processus, SOA et urbanisation du SI.

2) Les concepts liés aux processus métiers et aux SOA

- Introduction aux concepts des SOA.
- Les standards liés aux processus métiers : BPMN et BPEL, comparaison avec l'approche workflow classique.
- Structure et composants d'un SI orienté services : orchestrateur BPEL, référentiel de services, supervision BAM et SAM.

3) Introduction à l'approche méthodologique

- La modélisation métier (business modeling), implication de la MOA et adhésion à la démarche.
- Approche méthodologique : PRAXEME.
- Introduction à l'urbanisation du SI.

4) BPMN : Business Process Modeling Notation

- Introduction aux concepts BPMN, évolution de BPMN 1.0 et 2.0.
- Les acteurs des processus : acteurs externes et internes, gestion des habilitations.
- Swimlanes, pools, lanes.
- Activités et Tâches, Evénements.
- Fiche type de description des processus et activités.
- Connecteurs, flot de séquences et de message, groupes d'éléments.
- Les patterns liés aux workflows.
- Association, artefacts, annotations textuelles, objets métier échangés (modèle pivot).
- Comparaison BPMN et UML 2.0.
- Analyse et conception des processus, optimisation.
- Présentation de l'étude de cas.

5) Le modèle des objets métiers

- Introduction à la modélisation des objets métiers avec UML. Cycle de vie des objets.
- Lien avec les processus métiers : processus naturel lié au cycle de vie des objets métiers, l'approche PRAXEME.
- Développement d'exercices sur l'étude de cas.

6) Implémenter un processus avec BPEL

- Introduction à BPEL, évolution de BPEL.
- Liaison BPMN vers BPEL. Simulation des processus et génération du code BPEL.
- Mise en place d'applications composites.
- Piloter et superviser les processus : le BAM (Business Activity Management).

7) Panorama des outils

- Présentation de l'offre commerciale et Open Source.

Stage pratique de 3 jour(s)
Réf : AML

Participants

Analystes MOA ou MOE
amenés à intervenir dans des
projets SOA.

Pré-requis

Connaissances du formalisme
UML et des SOA.

Prix 2012 : 1730€ HT

Eligible DIF

Dates des sessions

Paris

10 sep. 2012, 3 déc. 2012

Modéliser une architecture SOA avec SoaML

OBJECTIFS

SoaML est une extension du formalisme UML adaptée pour les architectures orientées services. Ce stage apprendra aux analystes MOA et MOE à maîtriser les concepts de SoaML, à manipuler les différents éléments du langage, ceci afin de modéliser les processus métiers de l'entreprise et les services de leur architecture.

1) Architectures Orientées Services

2) Qu'est-ce que le standard SoaML ?

3) Le formalisme UML et l'approche MDA

4) La méthodologie SoaML

5) Définir le modèle Métier

6) Compléter le modèle métier

7) Définir le modèle Système

1) Architectures Orientées Services

- Spécificités des architectures SOA.
- Les différents constituants (Web Services, ESB, XML, BPEL, ...).
- La notion de service et le concept d'orchestration de services.
- La vision induite par la mise en place d'une SOA.
- L'alignement du SI sur les Processus Métier de l'organisation.
- L'impact sur le fonctionnement de la MOA et les relations MOA / MOE.

2) Qu'est-ce que le standard SoaML ?

- L'approche SOA dans les entreprises.
- Historique et objectifs de SoaML.
- Description de SoaML.
- Les besoins auxquels répondent ce standard.

3) Le formalisme UML et l'approche MDA

- Rappels sur les principaux diagrammes UML.
- Qu'est-ce que l'approche MDA (Model Driven Approach) ?
- Le lien entre UML et l'approche MDA.

4) La méthodologie SoaML

- Présentation générale de la méthodologie SoaML.
- L'étape de définition du modèle métier : Business Architecture Model.
- L'étape de définition du modèle système : System Architecture Model.

Travaux pratiques

Présentation de l'étude de cas, fil conducteur de la formation.

5) Définir le modèle Métier

- Description et objectifs du modèle d'architecture métier " Services Architecture ".
- Le concept de " Service Contract ".
- Le concept de " Participant ".

Travaux pratiques

Définition du modèle métier de l'étude de cas. Description des participants et des " services contracts ".

6) Compléter le modèle métier

- Modélisation des informations métiers.
- Modélisation des services : les interfaces de services et les différents types de messages.
- Les " chorégraphies " : description du comportement des services.

Travaux pratiques

Ajout à l'étude de cas des informations métiers, de la description des interfaces de services et des types de messages. Décrire le comportement des services avec un diagramme de séquence et d'activité.

7) Définir le modèle Système

- Description et objectifs du modèle d'architecture système.
- Spécifications des composants métiers.
- Définition des points d'accès aux services.

Travaux pratiques

Définition du modèle système dans le cadre de l'étude de cas. Description des composants métiers.

Stage pratique de 5 jour(s)
Réf : SOD

Participants

Ce cours est destiné aux développeurs et chefs de projets amenés à mettre en place une architecture SOA, et plus particulièrement un bus ESB à l'aide d'outils Open Source basés sur des standards issus du monde Java.

Pré-requis

Bonnes connaissances de la programmation Java, des technologies XML et des Web Services. Expérience requise en développement d'applications Java.

Prix 2012 : 2315€ HT

Eligible DIF

Dates des sessions

Paris

4 juin 2012, 24 sep. 2012
26 nov. 2012

SOA, mise en œuvre en Open Source/Java

OBJECTIFS

Ce cours vous permettra de comprendre les concepts de mise en œuvre d'une architecture SOA, à travers l'utilisation d'outils Open Source basés sur des standards tels que JBI, XML, SOAP, BPEL. Il met l'accent sur le traitement des messages entrants, les transformations de données, le routage intelligent et l'orchestration.

[1\) Introduction aux architectures SOA](#)

[2\) Principaux outils Open Source](#)

[3\) Les services Web](#)

[4\) Java Business Integration \(JBI\)](#)

[5\) Java Message Service](#)

[6\) Les transformations de données](#)

[7\) Le routage intelligent des messages](#)

[8\) L'orchestration des services](#)

1) Introduction aux architectures SOA

- Rappels sur les concepts et l'utilité des services Web.
- Les différents modèles d'intégration d'applications et les principes d'une architecture orientée services (SOA).
- Constituants d'un bus de messages ESB.
- Les notions de base des middlewares orientés messages (MOM), du routage intelligent des messages, des transformations XML, de l'orchestration avec BPEL.

2) Principaux outils Open Source

- Panorama. Fondation Apache. Communauté FUSE.
- L'offre JBossESB. Les projets OpenESB, MULE, PETAALS.
- Les moteurs BPEL : Apache ODE, ActiveBPEL, PXE...
- Les frameworks WebServices, tels que CXF ou Axis.
- Les MOM de type JMS, tels qu'ActiveMQ.

3) Les services Web

- Rappels sur le protocole SOAP, sur la description des services avec WSDL.

Travaux pratiques

Mise en œuvre d'un service Web simple avec Apache CXF ou Axis.

4) Java Business Integration (JBI)

- Les concepts de la spécification JBI (JSR 208).
- Le NMR (Normalized Message Router).
- Les liens entre JBI et WSDL. Les messages JBI.
- La notion de " Binding Components ".
- Les traitements au sein de JBI avec les " Services Engines " ou SE. Les SE dédiés BPEL, XSLT...
- Le packaging et le déploiement de composants JBI.

Travaux pratiques

Configuration de BC et SE JBI au sein de ServiceMix.

5) Java Message Service

- Le positionnement d'un MOM JMS dans une SOA.
- Introduction aux concepts JMS : types de destinations, formats de messages, envoi et réception...
- Le lien entre un ESB et un middleware JMS.

Travaux pratiques

Mise en œuvre à l'aide d'Apache ActiveMQ..

6) Les transformations de données

- L'utilité des transformations XML dans une infrastructure de type SOA. Possibilités.
- Introduction aux concepts de base XSLT, XQuery, XPath.

7) Le routage intelligent des messages

- Principes du routage basé sur le contenu.
- Le framework d'intégration Apache Camel.
- Configuration des règles de routage.
- L'intégration avec un ESB, un MOM JMS et une implémentation web services (type CXF).

Travaux pratiques

Mise en œuvre de différents types de routage.

8) L'orchestration des services

- Orchestration des services web. Langage WS-BPEL.

Travaux pratiques

Mise en œuvre d'une orchestration de services.

Stage pratique de 3 jour(s)
Réf : ODE

Participants

Chef de projets MOE,
architecte logiciel, concepteur
ou développeur.

Pré-requis

Bonnes connaissances de
l'architecture des systèmes
d'information et des Web
Services. Pratique de
la modélisation UML et
expérience en programmation
Java EE.

Prix 2012 : 1730€ HT

Eligible DIF

Dates des sessions

Paris

11 juin 2012, 8 oct. 2012
3 déc. 2012

Réaliser et déployer des processus métiers BPMN avec BPEL et Apache ODE

OBJECTIFS

Cette formation vous permettra de comprendre comment passer d'une modélisation BPMN dans un cadre SOA à la réalisation BPEL associée. Elle met l'accent sur les spécificités techniques BPEL et sur les différentes possibilités de réalisation. Vous apprendrez à titre de mise en oeuvre à déployer vos processus métiers au sein du moteur Apache ODE.

1) Les architectures SOA et la place des processus métiers

2) Les technologies indispensables

3) L'exploitation de modèles BPMN

4) Le langage BPEL

5) La mise en oeuvre avec Apache ODE

Travaux pratiques

Cette formation proposera de petits exercices unitaires pour assimiler les différentes notions, ainsi qu'une étude de cas couvrant les différentes activités étudiées : modélisation, réalisation, mise en oeuvre.

1) Les architectures SOA et la place des processus métiers

- Le modèle architectural SOA et son épine dorsale ESB.
- La modélisation des processus métiers et la place et l'usage des langages BPMN et BPEL. Alternatives

2) Les technologies indispensables

- Les technologies indispensables : XML et schéma XSD, transformations XSL, Web Services et WSDL, bus ESB, routage intelligent des messages.
- La place du moteur BPEL dans l'architecture SOA. Panorama des différents moteurs BPEL disponibles.
- Présentation d'Apache ODE.

3) L'exploitation de modèles BPMN

- Présentation du langage BPMN de l'OMG, comparaison avec les diagrammes d'activités UML.
- Les concepts BPMN : acteurs externes et internes, gestion des habilitations, activités et tâches, événements, fiche type de description des processus et activités, connecteurs, flot de séquences et de message, groupes d'éléments.

Travaux pratiques

Analyse d'un modèle BPMN.

4) Le langage BPEL

- Le lien entre processus BPEL et web services. L'impact sur WSDL.
- Les différents éléments du langage: les liens de partenariat, les variables, les séquences, les instructions. Echanges avec l'extérieur.
- Passage de BPMN à BPEL. Génération de code.
- Les concepts avancés de BPEL : parallélisme, synchronisations, solutions d'implémentation des activités humaines, mise en attente de données, gestion des erreurs, utilisation de web services tiers.
- Les difficultés d'implémentation liées à ces concepts avancés.

Travaux pratiques

Utilisation d'un outil graphique pour la génération de processus BPEL.

5) La mise en oeuvre avec Apache ODE

- Les différents moyens d'utiliser ODE. Les pré-requis d'installation.
- La création et le déploiement d'un processus.
- La gestion des versions.
- Les extensions BPEL. La gestion d'états. La gestion de la sécurité WS-Security.
- Le test des processus.
- L'administration

Travaux pratiques

Mise en oeuvre des processus produits au sein d'Apache ODE.

Stage pratique de 3 jour(s)
Réf : JBR

Participants

Ce stage est destiné aux développeurs, consultants, chefs de projets et analystes souhaitant maîtriser le processus de conception des règles avec JBoss Drools.

Pré-requis

Bonne connaissance du langage Java.

Prix 2012 : 1650€ HT

Eligible DIF

Dates des sessions

Paris

11 juin 2012, 8 oct. 2012
17 déc. 2012

JBoss Rules, mettre en oeuvre des règles métiers

OBJECTIFS

Ce cours vous apportera la connaissance nécessaire pour maîtriser l'évolution des règles métiers de votre système d'information. Vous apprendrez à mettre en place et à configurer le moteur de règles JBoss Rules (Drools). Vous définirez, déployerez et testerez des règles métiers au travers des différents modules proposés par cette plateforme.

1) Introduction

2) Présentation de JBoss Drools

3) Le langage de déclaration de règles

4) Moteur de règles Drools

5) Présentation détaillée des modules Drools

1) Introduction

- Les fondements de l'architecture SOA.
- Les éléments d'une SOA : ESB, Web Service, processus métier, orchestration des services.
- Méta-données et règles métiers.
- Le rôle des moteurs de règles.
- Les solutions disponibles sur le marché.

2) Présentation de JBoss Drools

- Historiques, définition, concepts.
- Les nouveautés de la version Drools 5.x.
- Les modules de JBoss Drools : Guvnor (BRMS/BPMS), Expert (Rules), Fusion (CEP), Flow (Process/Workflow) et Drools Planner.
- Drools API et Drools Core.
- Drools Server : Spring, Camel et CXF.
- JBoss Developer Studio, Drools IDE.

Travaux pratiques

Installation de la solution JBoss Drools. Mise en place d'Eclipse et Drools IDE.

3) Le langage de déclaration de règles

- Déclaration de règles et Drools Expert.
- Mots-clés, commentaires, package.
- Les fonctions.
- La déclaration de types.
- La déclaration de règles.
- La déclaration des requêtes (query).
- Contraintes, bindings, éléments conditionnels.
- Introduction à DSL (Domain Specific Language).
- Execution control et " rule flow ", Drools Flow.

Travaux pratiques

Déclaration des règles en utilisant Eclipse IDE.

4) Moteur de règles Drools

- La syntaxe XML. Quelle utilisation ?
- Présentation de Drools API.
- Introduction aux algorithmes RETE.
- Les sessions : Stateless/Stateful.
- Les agendas.
- Les phases : génération, déploiement et exécution.
- Paramètres de configuration de moteur.

Travaux pratiques

Déploiement des règles dans le moteur Drools.

5) Présentation détaillée des modules Drools

- L'administration et le test des règles avec Drools Governor (BRMS).
- La gestion d'événements complexes avec Drools Fusion.
- La planification automatique avec Drools Planner.
- La modélisation de workflows métier avec Drools Flow.

Travaux pratiques

Administration et test des règles avec BRMS. Démonstration des différents modules.

Stage pratique de 5 jour(s)
Réf : ZIB

Participants

Développeurs et architectes.

Pré-requis

Bonnes connaissances de la programmation Objet.
Connaissances de base du SQL, du XML et du Framework .NET. Expérience requise du langage C# ou VB.NET.

Prix 2012 : 2630€ HT

Eligible DIF

Dates des sessions

Paris

18 juin 2012, 10 sep. 2012
12 nov. 2012

Biztalk 2009, mise en oeuvre

OBJECTIFS

Ce cours intensif vous apportera toutes les connaissances pour concevoir, implémenter, déployer et tester des solutions d'intégration BizTalk Server 2009. Vous verrez son rôle et apprendrez à le mettre en oeuvre au sein d'une architecture SOA, à connecter divers systèmes (propriétaires, standards ou développés en .NET).

[1\) Introduction](#)

[2\) L'architecture](#)

[3\) Les Schémas](#)

[4\) Le Mapping](#)

[5\) Le routage de messages](#)

[6\) Les pipelines](#)

[7\) Les orchestrations](#)

[8\) Le déploiement](#)

[9\) Les adaptateurs et les ports](#)

[10\) Le moteur de règles](#)

[11\) Le monitoring avec BAM](#)

1) Introduction

- L'urbanisation du système d'information.
- Les concepts d'EAI et de SOA.

2) L'architecture

- Les composants de BizTalk Server.
- Les concepts et les outils.
- Les principes de fonctionnement.

3) Les Schémas

- Présentation des schémas.
- Les différents types.
- Les schémas XML et fichiers plats.

4) Le Mapping

- Introduction au Mapping.
- Comment mettre en place le Mapping.
- Configurer les " functoids " basiques et avancés.

Travaux pratiques

Exemple d'utilisation des schémas XML et des fichiers plats. Créer, valider et tester un Mapping. Exemples d'utilisation des " functoids ".

5) Le routage de messages

- Comment router les messages.
- L'architecture publish-subscribe.
- Les propriétés d'un message.

6) Les pipelines

- Le composant BizTalk pipeline.
- Envoyer et recevoir avec les pipelines.
- Les pipelines par défaut.
- Personnaliser les pipelines.

7) Les orchestrations

- Les principes et la gestion de l'orchestration.
- Les shapes.
- Corrélations, transactions et persistance.

Travaux pratiques

Mise en place du routage de messages. Configuration de port. Envoi et réception avec les pipelines. Mise en place d'une orchestration. Exemple de résolution de problèmes.

8) Le déploiement

- Préparation du déploiement. Utiliser Visual Studio et MSI. MSBuild : automatiser le déploiement.
- Gérer les versions.

9) Les adaptateurs et les ports

- Les adaptateurs de données, Line Of Business et WCF.
- Le Web Service et l'adapter SOAP.

10) Le moteur de règles

- Constituer une règle métier. Scénarios d'utilisation.
- Les scénarios d'utilisation.
- Les règles et l'orchestration.

Travaux pratiques

Exemple de déploiement. Utiliser les adaptateurs et les ports. Créer et déployer des règles métiers.

11) Le monitoring avec BAM

- Présentation du Business Activity Monitoring (BAM).
- Tableaux de bord. Indicateurs de performances.

Stage pratique de 5 jour(s)
Réf : DSJ

Participants

Ce cours est destiné aux développeurs et chefs de projets amenés à mettre en place des Web Services au moyen du langage Java.

Pré-requis

Bonnes connaissances du langage Java. Connaissances de base en XML. Expérience requise en développement Java.

Prix 2012 : 2315€ HT

Eligible DIF

Dates des sessions

Paris

21 mai 2012, 25 juin 2012
23 juil. 2012, 27 août 2012
24 sep. 2012, 22 oct. 2012
19 nov. 2012, 17 déc. 2012

Aix

15 oct. 2012, 10 déc. 2012

Bordeaux

10 sep. 2012, 19 nov. 2012

Bruxelles

10 sep. 2012, 19 nov. 2012

Genève

10 sep. 2012, 19 nov. 2012

Lille

11 juin 2012, 10 sep. 2012
19 nov. 2012

Luxembourg

10 sep. 2012, 19 nov. 2012

Lyon

15 oct. 2012, 10 déc. 2012

Nantes

8 oct. 2012, 3 déc. 2012

Rennes

8 oct. 2012, 3 déc. 2012

Sophia-antipolis

15 oct. 2012, 10 déc. 2012

Strasbourg

10 sep. 2012, 19 nov. 2012

Toulouse

10 sep. 2012, 19 nov. 2012

Développer des Web services en Java

OBJECTIFS

Ce cours vous permettra de comprendre l'intérêt et les enjeux des Web Services pour l'intégration d'applications (EAI). Il met l'accent sur les mécanismes de communication XML, SOAP et REST. Il présente les concepts de base et montre comment créer, déployer, mettre à disposition et utiliser des Services Web par l'emploi de JAX-WS (SOAP) et JAX-RS (REST). De nombreux exemples mettent en avant l'utilité des Services Web dans des contextes aussi variés que le Web, le B2C, le A2A et le B2B.

1) Introduction aux Web Services

2) Rappel sur XML et les API Java

3) Technologies Web Services SOAP/WSDL et REST

4) Le langage de description des Web Services (WSDL)

5) API Open Source pour la mise en place des Web Services SOAP/WSDL et REST

6) Publication et recherche de Web Services

7) Projets Open Source CXF, JAX-WS RI et Axis

8) Projets Open Source Jersey pour les services REST

9) Sécurisation des services Web

Travaux pratiques

Définition, développement, déploiement, configuration et sécurisation de Web Services. Développement de différents types de clients (lourd, Web, mobile, etc.) pour l'accès aux Web services. Un exercice global permettra de mettre en place une architecture complète basée sur les Web Services simulant des échanges de données applicatives entre des clients et un fournisseur de matériel informatique ainsi que des appels de méthodes distantes.

1) Introduction aux Web Services

- Définir le rôle des Web Services et l'intérêt varié de leur utilisation.
- Rappel sur le contexte des architectures n-tiers, applications et composants distribués.
- Approche Web Services, qu'est-ce qu'un Web Service, objectif, principe.
- Web services et EAI, A2A, B2C, B2B.
- Principales technologies des Web Services (SOAP, WSDL, UDDI).
- Web Services de type REST.
- Architecture orientée service (SOA).
- Le rôle de XML et de la plate-forme Java.
- Exemples mettant en oeuvre des Web Services.

Exercice

Exemples d'applications, comment invoquer un Web Service.

2) Rappel sur XML et les API Java

- Place du XML par rapport aux Web Services.
- Les langages à balises, domaines d'application de XML, le concept de document, objectifs.
- Echange de données, XML et SOAP.
- Besoins pour la création de requêtes SOAP et la manipulation des réponses.
- Les modèles de données : XML Schema.
- Les parseurs XML (API SAX et DOM), autres API.

Exercice

Comment manipuler des données XML dans une application Java via les APIs SAX, DOM et JAXB. Comment traiter les informations XML échangées via le protocole SOAP.

3) Technologies Web Services SOAP/WSDL et REST

- Description du protocole SOAP pour comprendre les mécanismes d'interopérabilité des Web Services.
- De XML à SOAP, anatomie d'un message SOAP, enveloppe SOAP.
- Gestion des erreurs avec SOAP Fault.
- Transmission de contenus binaires MTOM.
- Le langage de description des Web Services, WSDL (Web Services Description Language).
- Référencement de Web Services et annuaires, les solutions : UDDI, WS-Inspection, solutions de gouvernance OSS (Mule Galaxy)...
- Utilisation, architecture, rôle pour les services Web, publication et localisation, sécurité.
- Principes de REST, comment les mettre en oeuvre.

Exercice

Création de requêtes SOAP, envoi de requêtes SOAP. Interprétation des réponses. Déploiement d'un Web Service.

4) Le langage de description des Web Services (WSDL)

- Comprendre comment décrire un Web Service en XML.

- Introduction à WSDL.
- Anatomie d'un document WSDL.
- WSDL et Java.

Exercice

Création d'un fichier WSDL décrivant un Web Service.

5) API Open Source pour la mise en place des Web Services SOAP/WSDL et REST

- Les outils de programmation pour la création et l'utilisation de Web Services.
- Construire des Web Services et des clients avec JAX-WS. types supportés par JAX-WS et JAX-RS.
- Outils wsgen.
- Projet Open Source SOAP du groupe Apache.
- Utilisation de SOAP dans des contextes RPC.
- Le Web Services depuis des terminaux mobiles. Possibilités et restrictions.

Exercice

Programmation de Web Services de type échange SOAP/WSDL et REST. Utilisation des API Java JAX-WS et JAX-RS.

6) Publication et recherche de Web Services

- Comprendre le rôle des annuaires dans les applications mettant en oeuvre des Web Services. Rôle et limites d'UDDI.
- Comprendre la différence entre annuaires runtime et annuaires design-time.
- Comprendre comment mettre à disposition des Web Services via des annuaires UDDI.
- Programmation UDDI, structure de données UDDI.
- Vue d'ensemble et architecture de JAXR.
- Création et exécution d'un client JAXR, accès au registre, interrogation, gestion des données.
- Utilisation de descriptions WSDL avec UDDI.

Exercice

Publication d'un fichier d'un Web Service sur un annuaire UDDI. Recherche de Web Services.

7) Projets Open Source CXF, JAX-WS RI et Axis

- Comprendre les mécanismes des principales boîtes à outils de nouvelle génération SOAP.
- Architecture et composants des projets, CXF, JAX-WS RI et Axis.
- Installation, configuration, déploiement de services SOAP/WSDL.
- Consommer et publier des Web Services, correspondance données XML/Java.
- Utilisation des annotations spécifiques à JAX-WS (JSR 224) et des annotations pour le mapping Java/XML.
- Au moyen de Java : obtention du WSDL d'un service déployé et construction d'un WSDL.
- Outils : TCPMon, SOAP Monitor, SOAP UI version OSS.

Exercice

Mise en place d'une solution complète de communication entre applications avec SOAP/WSDL grâce à une de ces implémentations.

8) Projets Open Source Jersey pour les services REST

- Comprendre les mécanismes de mapping entre Java et le monde REST.
- Architecture et composants du projet Jersey.
- Installation, configuration, déploiement de services REST.
- Consommer et publier des Web Services REST.
- Utilisation des annotations spécifiques à JAX-RS (JSR 311) et des annotations pour le mapping Java/XML.
- Comment utiliser JAX-RS et Jersey pour publier des Services REST. Exemples de mise en oeuvre. Implémentations alternatives à Jersey (Restlet, ...).
- Outils : TCPMon, SOAP Monitor, SOAP UI version OSS.

Exercice

Mise en place d'une solution complète de communication entre applications avec une philosophie REST grâce à Jersey.

9) Sécurisation des services Web

- Authentification, autorisation, cryptage.
- Signatures digitales XML, cryptage XML des informations, toolkit Java, gestion des clés.
- Authentification personnalisée : les en-têtes SOAP.
- Extension de sécurité SOAP.
- Web Services Security Specifications (WS-Security).

Exercice

Sécurisation de l'application globale.

Stage pratique de 2 jour(s)
Réf : WER

Participants

Développeurs, chefs de projets.

Pré-requis

Connaissance de base en architecture et des applications Web. Bonnes connaissances de XML. Très bonnes connaissances en programmation Java. Expérience requise en développement JAX-RS et JAXB.

Prix 2012 : 1285€ HT

Eligible DIF

Dates des sessions

Paris

21 juin 2012, 4 oct. 2012
6 déc. 2012

Web Services RESTful, développement avancé en Java

OBJECTIFS

Cette formation vous permettra de compléter vos connaissances REST et de développer en Java avec JAX-RS des solutions complètes de type RESTful. Le cours présente notamment la spécification WADL, JSON et OAuth.

1) Rappels sur REST et JAX-RS

2) Invocation d'un service RESTful en Java

3) Description d'un service REST avec WADL ou WSDL

4) JavaScript Object Notation

5) Sécurisation d'un service REST

6) Eléments d'architecture

1) Rappels sur REST et JAX-RS

- Rappels sur les concepts Representational State Transfer et architecture RESTful.
- Rappels sur l'API JAX-RS.
- Rappels XML et JAXB.

Travaux pratiques

Installation de Jersey, développement d'une classe ressource. Développement d'un service complet.

2) Invocation d'un service RESTful en Java

- Utiliser l'API java de base pour récupérer une ressource.
- Utiliser l'API cliente de Jersey pour invoquer un service RESTful.
- Jakarta Commons HTTP Client.

Travaux pratiques

Invocation de services RESTful par diverses méthodes en Java.

3) Description d'un service REST avec WADL ou WSDL

- Présentation de la spécification Web Application description Language (WADL).
- Positionnement par rapport à WSDL 2.0, utilisable avec des services REST.

Travaux pratiques

Analyse d'une description WADL de services RESTful. Utilisation de l'outil wadl2java.

4) JavaScript Object Notation

- Intérêt des flux JSON dans les architectures RESTful.
- Manipulation de données JSON en Java.

Travaux pratiques

Développement d'un service RESTful retournant un flux JSON. Invocation du service et parsing du résultat en Java.

5) Sécurisation d'un service REST

- Rappels des possibilités d'action au niveau transport HTTP.
- Présentation du protocole OAuth permettant une authentification sur Internet, utilisé par Google, Flickr, Yahoo ...
- Mise en oeuvre avec JAX-RS.

Travaux pratiques

Sécurisation de services REST avec OpenSSO, OAuth et JAX-RS.

6) Eléments d'architecture

- Les différentes possibilités d'emploi de REST dans une architecture ouverte.

Stage pratique de 3 jour(s)
Réf : SEO

Participants

Ce cours est destiné aux développeurs et chefs de projets amenés à mettre en place des Web Services au moyen du langage Java.

Pré-requis

Bonnes connaissances du langage Java. Connaissances de base en XML. Expérience requise en développement JAX-WS.

Prix 2012 : 1730€ HT

Eligible DIF

Dates des sessions

Paris

4 juin 2012, 9 juil. 2012
10 sep. 2012, 12 nov. 2012

Web services SOAP, développement avancé en Java

OBJECTIFS

Ce cours détaille les fonctionnalités et API avancées des Web Services SOAP/JAX-WS ainsi que les extensions WS-ADDRESSING et WS-SECURITY. Il montre comment travailler au niveau du message SOAP, mettre en oeuvre des handlers côté client ou serveur, optimiser les flux XML et invoquer un service de façon asynchrone.

1) Rappels SOAP/WSDL et JAX-WS

2) Travail au niveau du message SOAP

3) Handlers SOAP et optimisation MTOM / XOP

4) Autres fonctionnalités

5) Sécurisation des services Web par WS-Security

Travaux pratiques

Mise en oeuvre des notions présentées, avec l'implémentation de référence proposée par GlassFish et quelques extensions.

1) Rappels SOAP/WSDL et JAX-WS

- Rappels sur les principales technologies des Web Services (SOAP, WSDL, UDDI).
- Rappels sur JAX-WS.
- Présentation des APIs étudiées.

Exercice

Analyse d'un document WSDL. Configuration de l'environnement de travail et réalisation d'un petit exercice de rappel de la mise en oeuvre de JAX-WS dans Eclipse.

2) Travail au niveau du message SOAP

- Pourquoi travailler au niveau du message ?
- Présentation de l'API Provider.
- Présentation de l'API Dispatch.
- Utiliser SAAJ pour construire et exploiter des messages SOAP.

Exercice

Comment manipuler des données XML dans une application Java via les APIs SAX, DOM et JAXB. Comment traiter les informations XML échangées via le protocole SOAP.

3) Handlers SOAP et optimisation MTOM / XOP

- Utilité d'un handler SOAP. Présentation du mécanisme.
- Création de handlers côté client et serveur, mise en place de chaînes de handlers.
- Optimisation des attachements binaires avec MTOM / XOP.

Exercice

Ajout de fonctionnalités à un service par mise en place de handlers. Optimisation du transfert d'images en attachement.

4) Autres fonctionnalités

- Invocation asynchrone d'un service.
- Utiliser JAX-RPC.
- Les concepts WS-ADDRESSING et leur mise en oeuvre.

Exercice

Invocation asynchrone d'un service et utilisation de WS-Addressing.

5) Sécurisation des services Web par WS-Security

- Présentation des concepts de l'extension WS-Security
- Authentification, autorisation, cryptage.
- Signatures digitales XML.
- Cryptage XML des informations. Gestion des clés.
- Authentification personnalisée : les en-têtes SOAP.
- Présentation de l'API JAAS.
- Politiques de sécurité avec WS-POLICY.

Exercice

Mise en oeuvre d'une authentification WS-SECURITY.

Stage pratique de 4 jour(s)
Réf : NCA

Participants

Développeurs .NET voulant appréhender tous les types de communication proposés par le framework.

Pré-requis

Très bonnes connaissances en programmation C#3 ou VB.NET avec Visual Studio 2005 ou 2008.

Prix 2012 : 2150€ HT

Eligible DIF

Dates des sessions

Paris

29 mai 2012, 25 sep. 2012
27 nov. 2012

.NET 3.5, communications entre applications Web Services, .NET remoting, WCF

OBJECTIFS

Ce cours vous permettra d'appréhender les possibilités de communication du framework .Net 3.5 et de choisir l'API la plus adaptée. Il traite des API bas niveau comme les Sockets jusqu'aux API haut niveau des services Web en passant par .Net Remoting et les WCF (Windows Communication Foundations).

1) Introduction

2) Les sockets

3) Les services Web ASP.NET

4) .NET Remoting

5) WCF

Travaux pratiques

Les exercices sont effectués en C# ou VB.Net suivant le choix des participants.

1) Introduction

- Présentation des différentes API.
- La réflexion pour les communications : rôle et présentation de l'API.
- La sérialisation pour les communications.

2) Les sockets

- Le protocole TCP/IP, communications mode stream et mode datagram.
- Traitement multithread à l'aide de Select et multithread à l'aide du pool de threads.
- Les NetworkStreams.

Travaux pratiques

Réalisation d'un serveur pouvant gérer la connexion simultanée de multiples clients.

3) Les services Web ASP.NET

- Notion de contrat WSDL et génération de proxies avec WSDL.exe et Visual Studio.
- Services Web avec état à l'aide de la Session.
- En-têtes SOAP.
- Authentification ASP.Net et application aux en-têtes SOAP.
- Les communications asynchrones et le cas particulier des pages ASP.NET.

Travaux pratiques

Réalisation d'un service Web accessible uniquement aux clients authentifiés. Gestion d'un contexte de client par un en-tête SOAP personnalisé. Réalisation d'une interrogation parallèle de multiples services Web de manière asynchrone.

4) .NET Remoting

- Objets sérialisés ou accessibles à distance. Notions de stub et de proxy. Types de canaux.
- Configuration par programmation / par fichier.
- Gestion de la durée de vie des objets par sponsors.
- Sécurité par cryptage et authentification des clients.
- Rôle des CallContexts.
- Problèmes de synchronisation liés aux Windows Forms.

Travaux pratiques

Réalisation d'un client et d'un serveur simple. Réalisation d'un serveur chat et de clients de type Windows Forms (gestion de problèmes de synchronisation).

5) WCF

- Principe et problématiques des architectures orientées services (SOA).
- Comprendre l'architecture WCF en vue d'unifier les techniques de communications.
- Créer et implémenter un contrat de service.
- Programmation et configuration des points de terminaison.
- Configuration et hébergement du service.
- Utilisation de sessions.
- Echanges synchrones et asynchrones.
- Mécanismes de sécurité et de transaction.

Travaux pratiques

Réalisation d'un client Windows et d'un serveur simple, avec différentes combinaisons de configuration des échanges côté client et côté serveur.