

les Cahiers  
du **Programmeur**

# J2EE

Avec la contribution  
de Stéphane **Bailliez**,  
Frédéric **Baudequin** et Gaël **Thomas**

© Groupe Eyrolles, 2003, 2005,

ISBN : 2-212-11574-1

**EYROLLES**



# Table des matières

---

<b>AVANT-PROPOS</b> .....	V
<b>INTRODUCTION À L'ÉTUDE DE CAS ET À J2EE</b> .....	1
BlueWeb : l'entreprise	2
L'équipe de développement	3
Technologies et méthodologie	4
L'application	6
Description des fonctionnalités	6
Analyse des données	6
Spécifications techniques	8
En résumé...	9
<b>1. UNE ARCHITECTURE À 5 COUCHES POUR BLUEWEB</b> .....	11
Un modèle à 5 couches pour le projet de gestion des signets	12
Couche de présentation des données : servlets	14
Objets métier – couche de persistance : EJB	15
Client riche – SWT	16
Déploiement	19
Base de données	20
En résumé...	21
<b>2. ENVIRONNEMENT DE DÉVELOPPEMENT CVS ET ANT</b> .....	23
Gestionnaire de code source : CVS	24
Production des versions : Ant	24
Ant – Vocabulaire de base	25
Adapter Ant à ses besoins	27
Tâches utilisateurs (Custom Tasks)	27
Codage en Java d'une tâche utilisateur	27
Principe de codage : convention JavaBean	27
Le build-file Ant et l'approche qualité	29
Ant : le couteau suisse du développement Java ?	29
Choix des bibliothèques	29
Critères de choix	30
BlueWeb joue la carte de l'Open Source	30
Le projet Jakarta Commons	31
Introduction : l'univers de la ligne de commandes	31
Le projet Commons CLI	34
Commons HTTP client	38
Commons Collections et Commons Primitives	43
En résumé...	45
<b>3. INTERFACES GRAPHIQUES POUR LA COUCHE CLIENTE</b> .....	47
Choix de la bibliothèque SWT	48
Retour à l'application de gestion des signets	49
Représentation graphique envisagée : l'arbre	49
Choix du widget : SWT ou JFace	50
Le TreeViewer JFace	50
Les acteurs	51
De l'utilité du découplage vues/données	52
Validation des saisies	53
Pourquoi valider des saisies ?	53
Comment vérifier une saisie ?	54
Maintenance du code : le framework JUnit	58
Les tests unitaires en pratique	60
Intégration des tests unitaires dans le cycle de vie du logiciel	62
Gestion des traces applicatives	64
Log4J ou java.util.logging ?	65
Log4J : concepts	66
En résumé...	71
<b>4. COUCHE DE PRÉSENTATION DES DONNÉES – SERVLETS HTTP</b> .....	73
Ouverture et maîtrise du couplage	74
Les servlets Java et HTTP	74
Rappels sur les servlets	74
Le pattern Commande et les requêtes HTTP	79
Schéma général du design de cette implémentation	79
Le pattern commande (Command) en quelques mots	80
Un design pattern en pratique : les Interceptors JBoss	81
Implémentation dans le projet BlueWeb	87
Présentation des données	89
Sérialisation d'objets Java	89
Une autre approche avec SOAP	96
En résumé...	97

<b>5. COUCHE MÉTIER AVEC LES EJB</b> .....	99
<b>Rappels sur les EJB 200</b>	
Un petit exemple de bean de session : un convertisseur d'euros 102	
Implémentation des services 102	
Home Interface 103	
Remote Interface (interface distante) 104	
Le design pattern Proxy 104	
Description du déploiement 108	
Programme client de test 111	
Conclusion partielle – travail nécessaire au codage d'un EJB à la main... 112	
<b>Introduction aux doclets 113</b>	
Manipuler les doclets 113	
<b>Outil rattaché : XDoclet 116</b>	
Introduction à XDoclet 116	
Ajoutez des doclets dans vos build-files 118	
Familiarisons-nous avec XDoclet 124	
<b>Intégrer XDoclet dans votre cycle de développement 127</b>	
Vers une amorce de solution 127	
AndroMDA : le candidat idéal 128	
Intégration de l'outil 129	
En résumé... 129	
<b>6. DÉPLOIEMENT ET GESTION DES VERSIONS</b>	
<b>AVEC ANT ET JAVA WEB START</b> .....	131
Gérer les versions et maintenir l'application 132	
Utilisation de Java Web Start sur le réseau BlueWeb 134	
Configuration du serveur web 134	
Création du fichier .jnlp 135	
Empaquetage de l'application : intégrer la signature de jar dans un build-file Ant 136	
Créer un trousseau de clés avec les outils du JDK 136	
Signer ses jars avec signjar 137	
Déploiement sur le serveur web 138	
Répercussions sur le code client pour l'équipe BlueWeb 138	
Rachel, un auxiliaire de choix 139	
Les design pattern en action : Adaptateur, Singleton et Factory 140	
En résumé... 143	
<b>7. AUDIT DU CODE ET QUALITÉ LOGICIELLE</b> .....	145
<b>Chartes et conventions de codage 146</b>	
L'outil Checkstyle 147	
Obtention d'un rapport avec Checkstyle 147	
<b>Utilisation de métriques 148</b>	
Exemple : le paquetage java.net 149	
Outil rattaché aux métriques : JDepend 149	
Comprendre les dépendances entre paquetages 150	
Utilisation des indicateurs de paquetages 151	
Utilisation de JDepend dans un build-file Ant 152	
<b>Tests unitaires : le framework JUnit 153</b>	
<b>Mise en forme du code source 154</b>	
Configurer votre propre convention 155	
Autres mises en forme 160	
Gestion des directives d'import de classes Java 160	
Analyse du code source 163	
Renforcement de l'architecture d'un projet 164	
Exemple concret chez BlueWeb 165	
Solution logicielle : Macker 166	
<b>Interaction avec CVS 169</b>	
Vers un build-file réaliste : rassemblons les morceaux... 171	
En résumé... 171	
<b>8. IMPLÉMENTATION DE LA LOGIQUE MÉTIER BLUEWEB</b>	
<b>AVEC XDOCKET</b> .....	173
Logique métier de l'application BlueWeb 174	
Code de l'EJB session (sans état) 174	
Code des entités (EJB entity type CMP) 179	
Génération du code et déploiement de nos EJB dans JBoss 185	
Configuration de PostgreSQL dans JBoss 188	
Contrôler la configuration de JBoss 191	
Tester nos composants 193	
Ce qu'il reste à faire pour la maquette BlueWeb 196	
En résumé... développer avec des outils libres 197	
<b>9. PREMIÈRES LEÇONS DU PROJET BLUEWEB</b> .....	199
<b>Les promesses des EJB 200</b>	
Un petit exemple 200	
Implémentation des services 200	
Composants et dépendances 202	
Injection de dépendances 202	
Ce que l'on peut espérer de ces produits 204	
Injection et instrumentation du code 204	
Struts ou un autre framework web ? 208	
La problématique 208	
Panel de produits envisageables 209	
Les frameworks MVC 209	
Technologies à base de modèles de documents 211	
Les inclassables 213	
Synthèse des solutions évoquées 215	
JMS 215	
Conclusion 216	
<b>INDEX</b> .....	217