SOME RIGHTS RESERVED

Acube, framework serveur

Prototype Spring Struts2 IBatis

Version x.y du JJ/MM/AAAA

Etat : xxx

SUIVI DES MODIFICATIONS

Version	Rédaction	Description	Vérification	Date
1.0	G.PICAVET	Initialisation		15/01/08
		Document validé dans sa version xxx		

LISTE DE DIFFUSION

Organisation	Nom	Info	Commentaire	Validation
		☐	☐	☐
		☐	☐	☐
		☐	☐	☐

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

SOMMAIRE

DOCUMENTS DE REFERENCE

Version	Titre

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

1 MOTIVATIONS

- Proposer une mise à jour de LISE moderne et en phase avec les standards du marché
- Améliorer la productivité lors du développement de la partie serveur JEE d'une application ACube.
- Limiter le couplage entre composants logiciel (pour une meilleure flexibilité de la solution).

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

2 INTERETS DE LA SOLUTION

2.1 STRUTS 2

Par rapport à la solution Struts 1.1 + Struts CX :

- dispose par défaut d'un générateur de flux xml via transformation xslt (en remplacement de StrutsCX)
- Les classes du modèle sont indépendantes du conteneur JEE (api Servlet) et simplifie donc les tests unitaires.
- Facilités de paramétrage applicatif (réutilisation par inclusion/ surcharge de paramétrage par défaut)
- pile d'intercepteurs de requête (permet d'empiler différent traitement sur une action, de manière paramétrable)
- dispose d'un plugin pour Spring

2.2 IBATIS

Par rapport à la solution JDBC Wrapper :

- Equivalent technique du JDBCWrapper
- Fonctionne avec toutes bases ayant un driver JDBC

database name	db version	driver vendor/name	driver version
Oracle	all	all	all, but recommend 10g
DB2	all	all	all
HSQLDB	all	n/a	n/a
MS SQL Server	2005	Sourceforge JTDS	1.1
MS SQL Server	2000	Sourceforge JTDS	1.0.3
MS SQL Server	2000	Microsoft	2.2.0040
MS SQL Server	2000	DataDirect	3.5
MySQL	4	Connector/J	3.1.8a
PostgreSQL	7/8	jdbc.PosgreSQL.org	7/8
Firebird	all	all	all
MSAccess	97	Sun JDBC-ODBC Bridge	bundled with JRE
MSAccess	2000	Sun JDBC-ODBC Bridge	bundled with JRE

Extrait de la page « http://opensource.atlassian.com/confluence/oss/display/IBATIS/Which+database-driver-combinations+are+known+to+work+with+iBATIS+DAO+and+SqlMap »

- Voir le Rex iBatis sur AdmiSource pour plus d'informations sur iBatis

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

2.3 SPRING & SPRING DAO

Permet de mieux architecturer l'application coté serveur, notamment grâce à :

- une fabrique de bean paramétrable et l'injection de dépendances.
- l'intégration de différents frameworks JEE (web-tiers et Business tiers), permettant de réduire le code et les dépendances.
- Gestion du transactionnel déclaratif (en remplacement du TransactionManager)

La partie Spring DAO présente un gain important pour les accès aux données.

Spring DAO incorpore des templates JDBC, iBatis ou Hibernate.
Cela réduit encore plus les efforts d'intégration d'iBatis dans une application web, notamment pour la gestion des transactions SQL.

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

3 APPLICATION PROTOTYPE

Prérequis : Tomcat 5.5.x + JRE 1.5

3.1 ARCHITECTURE

* Client riche : FRED v2.6

* Serveur JEE :

Utilise les frameworks : Spring 2.5, Struts 2.0.11 et Ibatis 2.3.0.

* Persistance :

La persistance est gérée par une base de données HSQL DB en mode In-process.

3.2 IMPLEMENTATION SERVEUR

3.2.1 CONFIGURATION

3.2.1.1 WEB-INF/WEB.XML

- chargement de Spring au démarrage du contexte. Fichiers de configuration : appContext.xml et daoContext-*.xml
- redirection des url /flux/protected/* vers le controller Struts2

3.2.1.2 WEB-INF/CLASSES/STRUTS.XML

- déclaration des mapping url/action, structurés en packages dérivant d'un super-package « acube-default »

3.2.1.3 WEB-INF/CLASSES/APPCONTEXT.XML

- definition des beans de Service à injecter dans les classes Action

3.2.1.4 WEB-INF/CLASSES/DAOCONTEXT-*.XML

- definition des beans DAO à injecter dans les classes de Service. Il existe une version bouchon (daoContext-bouchon.xml) et une version ibatis chargée par défaut (daoContext-ibatis.xml). La version ibatis définit la datasource à utiliser (ici une datasource simple sans pool de connexion), le mode de gestion des transactions et la localisation du fichier de mapping SQL

3.2.1.5 WEB-INF/CLASSES/SQL-HSQL.XML

- Recensement des fichiers de mapping ibatis utilisés par chaque package de l'application

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

3.2.1.6 WEB-INF/CLASSES/ACUBE/PROJECT/INTEGRATION/*/IBATIS/MAPS/*.XML

- requêtes SQL et mapping java utilisé pour un package

3.2.2 RENDU DES FLUX XML DYNAMIQUES

- struts sérialise automatiquement une action Struts en flux xml. Ce flux est ensuite transformé par une feuille xslt pour produire le flux demandé par le client.
- La génération du flux peut être effectuée par une JSP (un exemple est fournit sur l'action list)

3.2.3 VALIDATION SERVEUR

Démonstration de la validation déclarative de formulaire par Struts (menu form).

3.2.4PATTERN DELEGATE :

Le pattern Delegate a été remplacé par le pattern Service (ou Facade).

Justification : Le delegate servait essentiellement comme couche d'abstraction des DAO. Or cette abstraction est maintenant gérée par le framework Spring. Une couche de service métier est toutefois indispensable dans l'architecture d'une application. Celle-ci reste donc dans le package « business ».

3.2.5 GESTION DU TRANSACTIONNEL

Le prototype démontre la facilité de démarcation des transactions JDBC sans dépendance sur l'implémentation (JTA ou thread local). La démarcation est faite ici sur la couche service avec des annotations Spring, mais il est aussi possible de l'externaliser dans le fichier « appcontext.xml » avec des patterns sur nom de classe et méthode.

3.2.6 GESTION DES ERREURS

- la gestion des exceptions par Spring se fait sans rupture du principe d'encapsulation (pas d'exposition du choix d'implémentation dans les couches appelante avec un type d'exception par couche)

- Une exception lancée est interceptée par Struts2 puis la feuille ErreurVO.xsl transforme les données de l'exception selon son type (technique, fonctionnelle)

3.2.7 GESTION DE LA SECURITE

- contrôle d'habilitation simple via un intercepteur (droit d'accès d'un utilisateur à une action struts selon son role)

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne http://creativecommons.org/licenses/by-nc-sa/2.0/fr/ ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.