



Institut Supérieur d'Informatique
Modélisation et leurs Applications
Complexe des Cézeaux – BP 125
63173 AUBIERE CEDEX



Tutorial - Cours Java
3^e année F5

Framework Hibernate

Présenté par :
Guillaume CRESTA
GATCHA Charles
MOUNISSAMY Sivakumar

Responsable tutorial :
M. Cédric TESSIER

Octobre 2005



Institut Supérieur d'Informatique
Modélisation et leurs Applications
Complexe des Cézeaux – BP 125
63173 AUBIERE CEDEX



Tutorial - Cours Java *3^e année F5*

Framework Hibernate

Présenté par :
Guillaume CRESTA
GATCHA Charles
MOUNISSAMY Sivakumar

Responsable tutorial :
M. Cédric TESSIER

Octobre 2005

SOMMAIRE

SOMMAIRE

INTRODUCTION	4
I INTRODUCTION AUX FRAMEWORK	5
I.1 Définition d'un framework	5
I.2 Différents exemples de framework	5
I.2.1 <i>MFC</i>	5
I.2.2 <i>Jarka Struts</i>	6
I.2.3 <i>Spring</i>	6
I.3 Framework Hibernate	7
I.3.1 <i>Définition Hibernate</i>	7
I.3.2 <i>Architecture d'Hibernate</i>	8
I.3.3 <i>Comparatif avec Hibernate</i>	11
II APPLICATION WEB	13
II.1 Configuration d'Hibernate	13
II.2 Déclaration de l'objet Utilisateur	15
II.3 Servlet pour se connecter à une base de données	16
III Différents mapping objet/relationnel basiques	21
III.1 Déclaration de mapping	21
III.1.1 <i>Hibernate mapping</i>	22
III.1.2 <i>Class</i>	221
III.1.3 <i>Id (generator)</i>	25
III.1.4 <i>Discriminator</i>	26
III.1.5 <i>Property</i>	26
III.1.6 <i>One to one</i>	27
III.1.7 <i>Héritage</i>	28
III.1.8 <i>Many to one</i>	29
III.1.9 <i>Many to many</i>	30
III.1.10 <i>Component</i>	31
III.1.11 <i>Subclass</i>	32
III.1.12 <i>Import</i>	32
III.2 Types Hibernate	33
III.2.1 <i>Entités et valeurs</i>	33
III.2.2 <i>Les types de valeurs basiques</i>	33
III.2.3 <i>Type persistant d'énumération</i>	34
III.3 Identificateur SQL mis entre guillemets	34
IV Exemples	36
IV.1 Employeurs / employés	36
IV.2 Auteur / travail	38
CONCLUSION	41
OUVRAGES CONSULTES	42

INTRODUCTION

Travailler dans les deux univers que sont l'orienté objet et la base de données relationnelle peut être lourd et consommateur en temps dans le monde de l'entreprise d'aujourd'hui. Un outil de mapping objet/relationnel pour le monde de java a donc été mise en place : Hibernate. Le terme mapping objet/relationnel (ORM) décrit la technique consistant à faire le lien entre la représentation objet des données et sa représentation relationnelle basé sur un schéma SQL.

Suite à cela, de nombreuses versions d'Hibernate ont été mises à disposition des utilisateurs afin de le faire évoluer. Hibernate se base sur l'architecture Modèle – Vue – Contrôleur qui fut introduite comme partie du SmallTalk80 afin de limiter les efforts de programmation liés à l'élaboration de système. Hibernate est agnostique en terme d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd que dans un environnement web léger de type Tomcat (Apache) ou dans un environnement J2EE complet (Weblogic, Websphere, JBoss).

Notre tutorial s'organisera donc de la façon suivante. Nous verrons dans une première partie l'introduction aux frameworks. Dans un second temps, nous aborderons une application Web afin d'observer la configuration d'Hibernate et la déclaration des objets. Pour finir, nous développerons les différents mappings utiles suivis de quelques exemples avant d'apporter une rapide conclusion à ce tutorial.

I INTRODUCTION AUX FRAMEWORK

I.1 Définition d'un framework

Le framework désigne le cadre dans lequel va s'insérer une application. En programmation orientée objet, il désigne l'infrastructure logicielle qui facilite la conception des applications par l'utilisation de bibliothèques de classes ou de générateurs de programmes. C'est une bibliothèque de classes fournissant une ossature générale pour le développement d'une application dans un domaine particulier. Ces composants sont organisés afin d'être utilisés en interaction les uns avec les autres et sont spécifiques généralement à un type d'application.

Les frameworks facilitent ainsi le travail du développement en fournissant un squelette d'application qu'il suffit de remplir pour l'adapter à ses besoins. La contrepartie est qu'un framework représente un sur ensemble de tous les besoins génériques, ce qui conduit à supporter un grand nombre de choses même si souvent, une toute petite partie est utile pour le cas à réaliser. Ceci complique également la tâche d'apprentissage et d'assimilation de l'environnement par le développeur.

Un framework est un ensemble de classes abstraites qui, dans le but de faciliter la création d'une partie d'un système logiciel, collaborent entre elles. Il partage le domaine visé en classes abstraites et permet de définir les responsabilités de chacune par rapport aux autres et les rapports entre elles.

I.2 Différents exemples de framework

I.2.1 MFC

Les MFC de Microsoft est un exemple de framework qui permettent de développer une application en C++ basée sur une architecture *Fenêtre Cadre-Document-Vue*.

MFC est donc une bibliothèque de classes C++ fournissant un cadre général pour la programmation sous Windows. MFC encapsule une grande partie des fonctions API Windows et améliore la logique de la création des interfaces.

L'objet principal du MFC est donc de fournir un cadre général pour développer facilement des interfaces graphiques. Pour cela, il fournit une architecture de programme basée sur la notion de document et vue. Pour les MFC, une interface graphique est un moyen parmi d'autres de visualiser des données (document) ; cette visualisation (vue) s'effectuant dans une fenêtre cadre elle-même pilotée par un programme principal.

1.2.2 Jarka Struts

Struts est un exemple de framework open source, basé sur l'architecture MVC (Modèle Vue Contrôleur). C'est un projet d'un framework faisant partie de Apache Jarka Project. Il sert à développer des applications pour le Web.

Le cœur du framework Struts est une couche contrôleur basée sur les technologies les plus acceptées comme le JSP, le JavaBeans, et le XML. Struts encourage les architectures basées sur l'approche Model 2, qui est une variante du modèle classique MVC. Struts fournit son propre composant contrôleur et intègre d'autres technologies pour offrir le Modèle et la Vue. Pour le modèle, Struts peut interagir avec toutes les technologies d'accès aux données comme les EJB, le JDBC. Pour la vue, Struts fonctionne bien avec les JSP, les Velocity Templates et d'autres systèmes de présentation.

1.2.3 Spring

Spring est un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'application J2EE. Il prend donc en charge la création d'objets et leurs mises en relation par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ceux-ci.

Le gros avantage par rapport aux serveurs d'application est qu'avec Spring, vos classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que Spring est qualifié de conteneur « léger ».