

# Spring

## Première Partie (mise en place)

- 1) Récupérer les archives nécessaires pour l'utilisation de Spring (voir la partie Requirements / Spring)
- 2) Dans le cas d'un projet JAVA (pas un projet Web JSP/Servlet). Assurez-vous que votre projet pointe sur les archives suivantes :
  - **spring.jar** qui se trouve dans le répertoire « dist » du framework spring (quand vous aurez dézipper l'archive Spring en 1))
  - L'archive **spring-aop.jar** qui se trouve dans le répertoire « dist/modules »
  - toutes **les archives** qui se trouvent dans « \lib\aoalliance » et dans « \lib\aspectj »

**Important : dans le cas d'un projet Web. Assurez-vous d'avoir ces archives dans le répertoire WEB-INF/lib**

- 3) Assurez-vous si dans votre projet vous utilisez Hibernate de remplir les requirements décrites ci-dessous (partie pour Hibernate)

## Deuxième Partie (Inversion of Control)

Dans cette partie il s'agit d'appliquer l'inversion de control (IoC) de Spring. En reprenant les notes du cours essayez dans un premier temps de :

- Créer votre premier fichier de configuration Spring pour la définition des Beans. Ce fichier devra s'appeler applicationContext.xml. Dans le cas d'un projet Java, ce fichier doit être à la racine du projet. **Dans le cadre d'un projet Web, ce fichier doit être dans WEB-INF (i.e. WEB-INF/applicationContext.xml). Aussi dans le cadre d'un projet Web vous devrez rajouter des lignes dans le web.xml afin de prendre en compte Spring. Ces lignes concernent la configuration des Listners (voir diapos cours)**
- Dans ce fichier vous devez définir votre premier Bean. Ca sera le Bean **DAOContact**. Cela implique qu'à l'intérieur de vos servlets/class java, vous n'instanciez plus directement DAOContact mais que vous demandiez à Spring de le faire à votre place. Faites-le pour tous les autres DAO si nécessaire.
- Testez maintenant la création de Beans avec les deux méthodes i.e. par constructeur et par propriétés. Cela se fera dans le contexte de la création de deux contacts mais cette fois-ci pas à l'intérieur de votre application web/java (à travers le formulaire ajout d'un nouveau contact ou bien d'une fenêtre swing Java) mais par l'utilisation du conteneur de Spring. Dans la servlet/class d'ajout d'un nouveau contact vous aurez donc à appeler le bean pour le DAOContact, et deux Beans pour Contact.
- Assurez-vous que l'instance du DAO se charge bien de sauvegarder ces deux instances de Contact dans la base de données (toujours en utilisant le support d'hibernate)
- Faites la même chose pour les autres fonctionnalités (Servlets/ classes de suppression, de recherche, etc.)

## Troisième Partie (Aspects)

Les aspects sont l'un des apports les plus importants de Spring. Dans cette partie on vous demande de les tester/pratiquer.

Pour cela vous devrez prendre des opérations au choix dans vos POJOs (dans Contact par exemple) et essayer de tisser des aspects au niveau de ces classes i.e. dévier le comportement par défaut d'une opération ou deux de vos POJOs. Pour cela basez-vous

sur les notes de cours. Pour ce faire, vous devrez modifier le fichier XML et peut être créer des classes pour les aspects.

#### **Quatrième Partie (Intégration Hibernate en utilisant le Template Hibernate)**

Cette partie est un peu difficile et demande pas mal de travail mais elle vous aidera à comprendre pas mal de choses au niveau de Spring, notamment l'utilisation des Transactions Hibernate à travers un tissage d'aspect, l'injection des dépendances, le bean datasource, etc. Pour ce faire, vous devez :

- Dans un premier temps, enlevez tout le code propre à Hibernate de vos classes /servlet (i.e. Session Factory, création de la session, les transactions, les exceptions etc.).
- Ensuite, en suivant les notes du cours vous devrez déclarer dans le fichier de configuration applicationContext.xml tous les aspects liés à la data source (nom base, driver, pass et user, etc.) qui sont pour le moment définis dans hibernate.cfg.xml
- Toujours en suivant les notes du cours, définissez les beans pour la Session Factory et pour les transactions. Rajouter ensuite l'aspect qui gère le tissage d'aspect des transactions au niveau du template hibernate et principalement au niveau des méthodes save, find, delete, etc.
- Finalement, modifiez votre code pour utiliser maintenant le Template Hibernate pour Spring dans votre application.
- Tester maintenant vos modifications et assurez-vous que ça marche !

#### **Requirements Environnement Servlet/JSP + Hibernate:**

##### **Pour Spring**

Télécharger le framework Spring à partir de :

<http://www.springsource.com/download?project=Spring%20Framework>.

Choisissez la Community Version. Prenez la version « with dependencies ». Déziper l'archive quelque part dans votre machine.

Télécharger Spring IDE à partir de <http://springide.org/blog/2008/09/30/spring-ide-22-released/> et intégrez-le avec Eclipse (soit en utilisant le update manager soit le zip)

Vous devez aussi avoir les Jars suivants soit dans votre build path soit dans le « lib » dans le cas d'un web project : commons-collections-3.1.jar, commons-dbcp.jar, commons-logging.jar, commons-pool.jar (vous pourrez les trouver sur le site DocJar.com)

**SINON vous pouvez récupérer toutes les lib nécessaires à l'exécution de Spring + Hibernate + MySQL en cliquant sur ce lien :**

[http://pagesperso-systeme.lip6.fr/Reda.Bendraou/mdoc/Spring-Hibernate-Mysql\\_lib.zip](http://pagesperso-systeme.lip6.fr/Reda.Bendraou/mdoc/Spring-Hibernate-Mysql_lib.zip)

##### **Pour Servlet/JSP**

Télécharger la dernière release d'Eclipse version JEE développeur à l'adresse suivante (ou plus sûr, eclipse.org -> downloads->Eclipse for JEE dev.) :

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/releases/ganymede/SR2/eclipse-jee-ganymede-SR2-win32.zip> Cette version doit contenir WTP

Télécharger la dernière version du JDK sur : <http://developers.sun.com/downloads/>

Après installation (de préférence directement sur C: ou bien C:\MonRep\JDK1.X), assurez vous d'avoir une variable d'environnement JAVA\_HOME qui pointe vers le répertoire d'installation du JDK.

Rajoutez à la variable PATH de votre environnement l'entrée suivante : %JAVA\_HOME%\bin

Télécharger la dernière version de Tomcat et déziper là quelque part sur votre disque. Avoir un c:/tomcat6.0/ serait pas mal.

Tester Tomcat en lançant le serveur à partir de son répertoire bin puis sur firefox, tapez <http://www.localhost:8080/>

La fenêtre Tomcat est sensée s'afficher. Pour l'administration allez sur Tomcat Manager

Télécharger la dernière version de MySQL database server Sur : <http://dev.mysql.com/downloads/>

Pour pouvoir y accéder partout depuis une fenêtre Dos, rajouter une entrée dans votre variable d'environnement PATH : c:\MonRepInstall\...\MySQL Server 6.0\bin;

Lors de l'installation faire attention au login/mot de pass à définir pour la base. Choisir par exemple root (login) et root (password) ou bien root(login) et (rien du tout) comme password

Télécharger la dernière version du driver MySQL pour Java sur :

<http://dev.mysql.com/downloads/connector/j/5.1.html>

Placez le .jar du connector JDBC dans le repertoire WebContent/WEB-INF de votre projet web

Finalement, bouton droit sur votre projet-> properties-> Java Build Path -> Libraries -> add External Jars -> puis pointer vers les servlet\*.jar et servlet\*jsp\*.jar qui se trouvent dans le répertoire plugin de votre eclipse.

## Pour Hibernate

Télécharger la dernière version stable d'Hibernate Tools for Eclipse sur:

<http://www.hibernate.org/6.html> (or hibernate.org-> downloads-> last release of hibernate tools)

Pour l'installer, juste déziper et copie/coller le contenu du package « plugin » à destination du package du même nom d'Eclipse. Même chose pour le répertoire « features »

Télécharger et déziper la dernière version stable d'hibernate framework, Hibernate – Core sur: <http://www.hibernate.org/6.html> (or hibernate.org-> downloads-> last stable release hibernate core)

**IMPORTANT:** Assurez-vous que votre projet pointe vers le fichier hibernate(+num version).jar (qui se trouve dans le répertoire Hibernate-Core que vous venez juste de déziper). Pour cela : (votre projet->B. Droit->properties->Java Build Path->add external Jars->parcourir et pointer vers le hibernate3.jar –version 3 dans notre cas). **En plus :** Assurez-vous que votre projet pointe vers **TOUS** les fichiers .jar qui se trouvent dans le répertoire lib/required du répertoire Hibernate Core.

**ATTENTION :** Si vous êtes dans le contexte d'un projet web i.e. Servlet/JSP (dans votre cas oui), vous pouvez sauter cette étape et à la place mettre tous les jars décrits précédemment dans le package webcontent/WEB-INF de votre projet Web.

Télécharger et déziper la dernière version stable de slf4j sur : <http://www.slf4j.org/download.html> hibernate utilise cette API pour les logs.

Assurez vous que votre projet pointe sur les jars suivants : slf4j-api-1.5.8.jar (il se trouvait déjà dans lib/required d'hibernate code mais ce n'est pas grave) et slf4j-simple-1.5.8.jar (votre projet->B. droit ->properties->Java Build Path->add external Jars->parcourir et pointer les 2 jars).

**ATTENTION** : Si vous êtes dans le contexte d'un projet web i.e. Servlet/JSP (dans votre cas oui), vous pouvez sauter cette étape et à la place mettre tous les jars décrits ci-dessus dans le package webcontent/WEB-INF de votre projet Web.

Optionnellement vous pouvez utiliser Toad pour visualiser votre base de données. Beaucoup plus pratique que la simple vue Dos.

<http://www.quest.com/toad-for-mysql/>

