

Spring

Première Partie (mise en place)

1) Récupérer les archives nécessaires pour l'utilisation de Spring (voir votre ZIP de TP, vous y trouverez les libs SPRING)

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

2) 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. **ATTENTION : utiliser le wizard de Spring IDE pour le faire. Si votre projet n'est pas reconnu comme étant un projet Spring, bouton droit sur votre projet->Spring tools->add spring capabilities.** 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

Vous avez une version de Spring IDE dans votre ZIP de Projet. Autrement, 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)

ATTENTION : il faut redémarrer Eclipse. Si avez déjà un projet java existant et que vous voulez faire du Spring, bouton droit sur votre projet->Spring tools->add spring capabilities

