

MT4 - Module Java - Travaux pratiques

Exercice 9

Ajouter Spring framework à votre projet avec comme objectif de lui laisser la charge de l'instanciation de votre composant de lecture.

- La librairie à intégrer est [la suivante](#)
- Créer un fichier *beans.xml* dans le dossier **src/main/resources** dans lequel vous allez déclarer un bean "reader"
- Dans votre programme, remplacer l'instanciation de votre composant de lecture (le cas échéant, l'appel à une factory) par l'appel de la méthode `getBean("reader")` ; sur le contexte Spring que vous aurez déclaré / initialisé au préalable.

Vérifier le bon fonctionnement de votre programme, puis modifier la définition du bean "reader" pour définir l'implémentation alternative (FileSystem vs. Base de données).

Exercice 10

S'assurer que le bean "reader" configure votre solution de lecture en base de données.

Etoffer votre configuration en externalisation la définition des paramètres d'accès à la base de données dans un fichier *application.properties* à positionner dans le dossier **src/main/resources** (cf. PJ dans HETIC Learn)

Ajuster votre configuration Spring (*beans.xml*) pour indiquer à Spring de prendre en compte le fichier *application.properties* et la déclaration du bean *reader* pour faire référence aux clés de configuration positionnées dans *application.properties*

Exercice 11

Créer une application Spring Boot *learning-spring* via <https://start.spring.io/> en choisissant les bons outils et bonnes versions, et en ajoutant aux niveau des dépendances (dans l'assistant) :

- JDBC API
- PostgreSQL Driver

Ajouter la configuration de la base de données dans le fichier *application.properties*

Créer une classe **FichierJDBCRepository** annotée **@Repository** avec un champ de type **JdbcTemplate** initialisé par le constructeur.

Implémenter l'équivalent de votre code de lecture produit à l'exercice 7 dans une méthode de **FichierJDBCRepository** en utilisant l'API **JdbcTemplate**.

Vérifier le bon fonctionnement du chargement des données en l'intégrant dans le *main* de votre application.

Exercice 12

Epurer la codebase *learning-java* pour retirer tout le code désormais implémenté dans *learning-spring*

Intégrer votre jar *learning-java* dans le projet *learning-spring* sous la forme d'une dépendance Maven

Ajuster le contenu du *main* pour reproduire le fonctionnement cible du programme (lecture > calculs > écriture) en s'appuyant sur les définitions existantes dans la désormais bibliothèque *learning-java* (et ceci sans dupliquer du code : si vous estimez devoir dupliquer du code, vous êtes censés le supprimer de *learning-java*).