



# Setup Environnement de développement

# Environnement de Développement

Mon conseil :  
Reprenez des fichiers  
Existants pour éviter  
les surprises

- Incorporez les fichiers de configuration web.xml , et spring-mvc-config-servlet.xml
- Incorporer les librairies JSTL
- Incorporer les librairies de Spring
- Incorporer les librairies d'hibernate

# Synthèse des dépendances

## Hibernate

FastInfoset-1.2.15.jar  
antlr-2.7.7.jar  
byte-buddy-1.10.10.jar  
c3p0-0.9.5.5.jar  
classmate-1.5.1.jar  
dom4j-2.1.3.jar  
hibernate-c3p0-5.4.20.Final.jar  
hibernate-commons-annotations-5.1.0.Final.jar  
hibernate-core-5.4.20.Final.jar  
istack-commons-runtime-3.0.7.jar  
jandex-2.1.3.Final.jar  
javassist-3.24.0-GA.jar  
javax.activation-api-1.2.0.jar  
javax.persistence-api-2.2.jar  
jaxb-api-2.3.1.jar  
jaxb-runtime-2.3.1.jar  
jboss-logging-3.3.2.Final.jar  
jboss-transaction-api\_1.2\_spec-1.1.1.Final.jar  
mchange-commons-java-0.2.19.jar  
stax-ex-1.8.jar  
txw2-2.3.1.jar

## Jstl

javax.servlet.jsp.jstl-1.2.1.jar  
javax.servlet.jsp.jstl-api-1.2.1.jar

## MySQL

mysql-connector-java-8.0.21.jar

## Spring

javax.annotation-api-1.3.2.jar  
spring-aop-5.2.2.RELEASE-javadoc.jar  
spring-aop-5.2.2.RELEASE-sources.jar  
spring-aop-5.2.2.RELEASE.jar  
spring-aspects-5.2.2.RELEASE-javadoc.jar  
spring-aspects-5.2.2.RELEASE-sources.jar  
spring-aspects-5.2.2.RELEASE.jar  
spring-beans-5.2.2.RELEASE-javadoc.jar  
spring-beans-5.2.2.RELEASE-sources.jar  
spring-beans-5.2.2.RELEASE.jar  
spring-context-5.2.2.RELEASE-javadoc.jar  
spring-context-5.2.2.RELEASE-sources.jar  
spring-context-5.2.2.RELEASE.jar

## Spring (suite)

spring-context-indexer-5.2.2.RELEASE-javadoc.jar  
spring-context-indexer-5.2.2.RELEASE-sources.jar  
spring-context-indexer-5.2.2.RELEASE.jar  
spring-context-support-5.2.2.RELEASE-javadoc.jar  
spring-context-support-5.2.2.RELEASE-sources.jar  
spring-context-support-5.2.2.RELEASE.jar  
spring-core-5.2.2.RELEASE-javadoc.jar  
spring-core-5.2.2.RELEASE-sources.jar  
spring-core-5.2.2.RELEASE.jar  
spring-expression-5.2.2.RELEASE-javadoc.jar  
spring-expression-5.2.2.RELEASE-sources.jar  
spring-expression-5.2.2.RELEASE.jar  
spring-instrument-5.2.2.RELEASE-javadoc.jar  
spring-instrument-5.2.2.RELEASE-sources.jar  
spring-instrument-5.2.2.RELEASE.jar  
spring-jcl-5.2.2.RELEASE-javadoc.jar  
spring-jcl-5.2.2.RELEASE-sources.jar  
spring-jcl-5.2.2.RELEASE.jar  
spring-jdbc-5.2.2.RELEASE-javadoc.jar  
spring-jdbc-5.2.2.RELEASE-sources.jar  
spring-jdbc-5.2.2.RELEASE.jar  
spring-jms-5.2.2.RELEASE-javadoc.jar  
spring-jms-5.2.2.RELEASE-sources.jar  
spring-jms-5.2.2.RELEASE.jar

## Spring (fin)

spring-messaging-5.2.2.RELEASE-javadoc.jar  
spring-messaging-5.2.2.RELEASE-sources.jar  
spring-messaging-5.2.2.RELEASE.jar  
spring-orm-5.2.2.RELEASE-javadoc.jar  
spring-orm-5.2.2.RELEASE-sources.jar  
spring-orm-5.2.2.RELEASE.jar  
spring-oxm-5.2.2.RELEASE-javadoc.jar  
spring-oxm-5.2.2.RELEASE-sources.jar  
spring-oxm-5.2.2.RELEASE.jar  
spring-test-5.2.2.RELEASE-javadoc.jar  
spring-test-5.2.2.RELEASE-sources.jar  
spring-test-5.2.2.RELEASE.jar  
spring-tx-5.2.2.RELEASE-javadoc.jar  
spring-tx-5.2.2.RELEASE-sources.jar  
spring-tx-5.2.2.RELEASE.jar  
spring-web-5.2.2.RELEASE-javadoc.jar  
spring-web-5.2.2.RELEASE-sources.jar  
spring-web-5.2.2.RELEASE.jar  
spring-webflux-5.2.2.RELEASE-javadoc.jar  
spring-webflux-5.2.2.RELEASE-sources.jar  
spring-webflux-5.2.2.RELEASE.jar  
spring-webmvc-5.2.2.RELEASE-javadoc.jar  
spring-webmvc-5.2.2.RELEASE-sources.jar  
spring-webmvc-5.2.2.RELEASE.jar  
spring-websocket-5.2.2.RELEASE-javadoc.jar  
spring-websocket-5.2.2.RELEASE-sources.jar  
spring-websocket-5.2.2.RELEASE.jar

# Si votre jvm > = Java 9

- Incorporez ces dépendances

`javax.activation-1.2.0.jar`

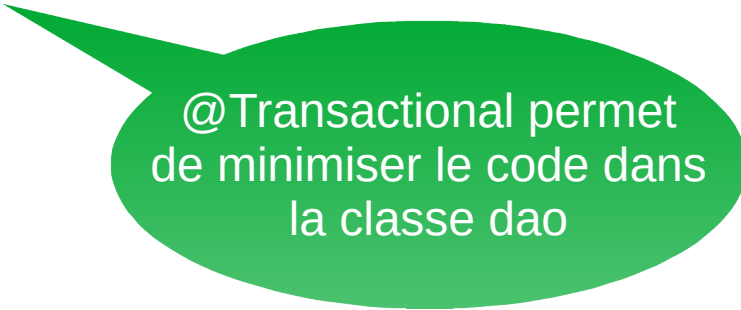
`jaxb-api-2.3.0.jar`

`jaxb-core-2.3.0.jar`

`jaxb-impl-2.3.0.jar`

# Setup (suite) : Hibernate

- Définir une bdd dataSource / connection pool
- Configurer la session factory hibernate
- Configurer le gestionnaire de transaction d'hibernate
- Activer la configuration des transaction par annotation



@Transactional permet de minimiser le code dans la classe dao

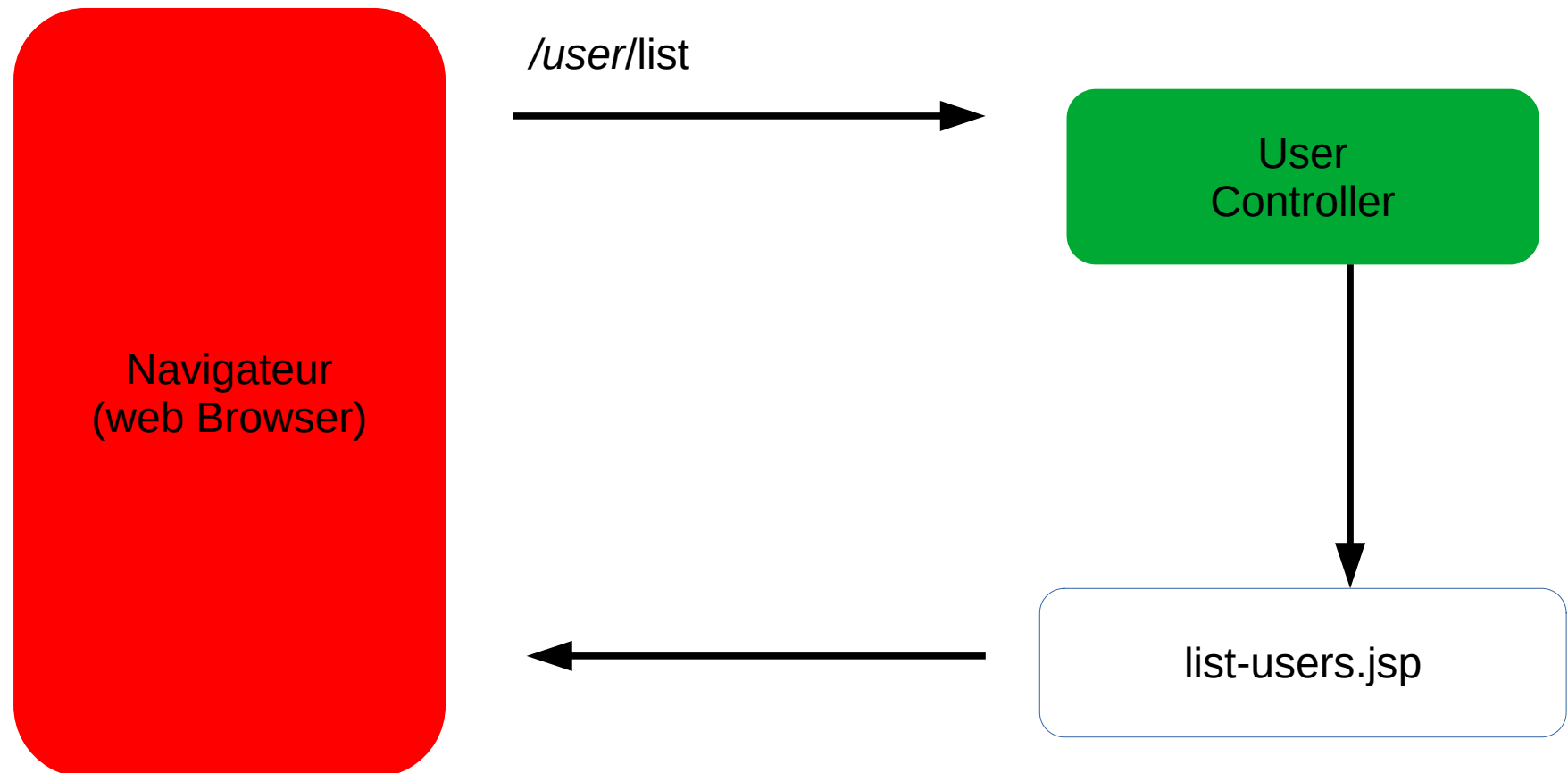
Cf spring-mvc-config-servlet.xml

# Spring-mvc-config-servlet.xml

- On retrouve les elements de configuration mvc déjà vus basePackage, view resolver , dispatcher servlet
- On peut relire ce fichier de configuration de spring comme une intégration des propriétés que l'on mettrait dans hibernate.cfg.xml

# On va pouvoir commencer à coder ...

- Tester le controlleur de spring MVC





# Créons ce premier cas d'utilisation

- Créer un package, `com.tld.controller`
- Puis une classe `UserController.java`
- Créer un sous dossier `WEB-INF/view`
- Puis y créer un fichier `list-users.jsp`

Exécutez l'application (Run on Server sur le dossier de l'application)

Puis entrez la fin de l'url dans la barre d'adresse pour envoyer la requête mappée dans notre contrôleur

# Test ok



Ici bientôt s'affichera une liste de users

Tout fonctionne de bout en bout sur notre premier use case de test. (end to end )