



Formation esup-commons

V1.3.9

Doriane Dusart, Jean-Marie Lemay
27, 28, 29 avril 2011

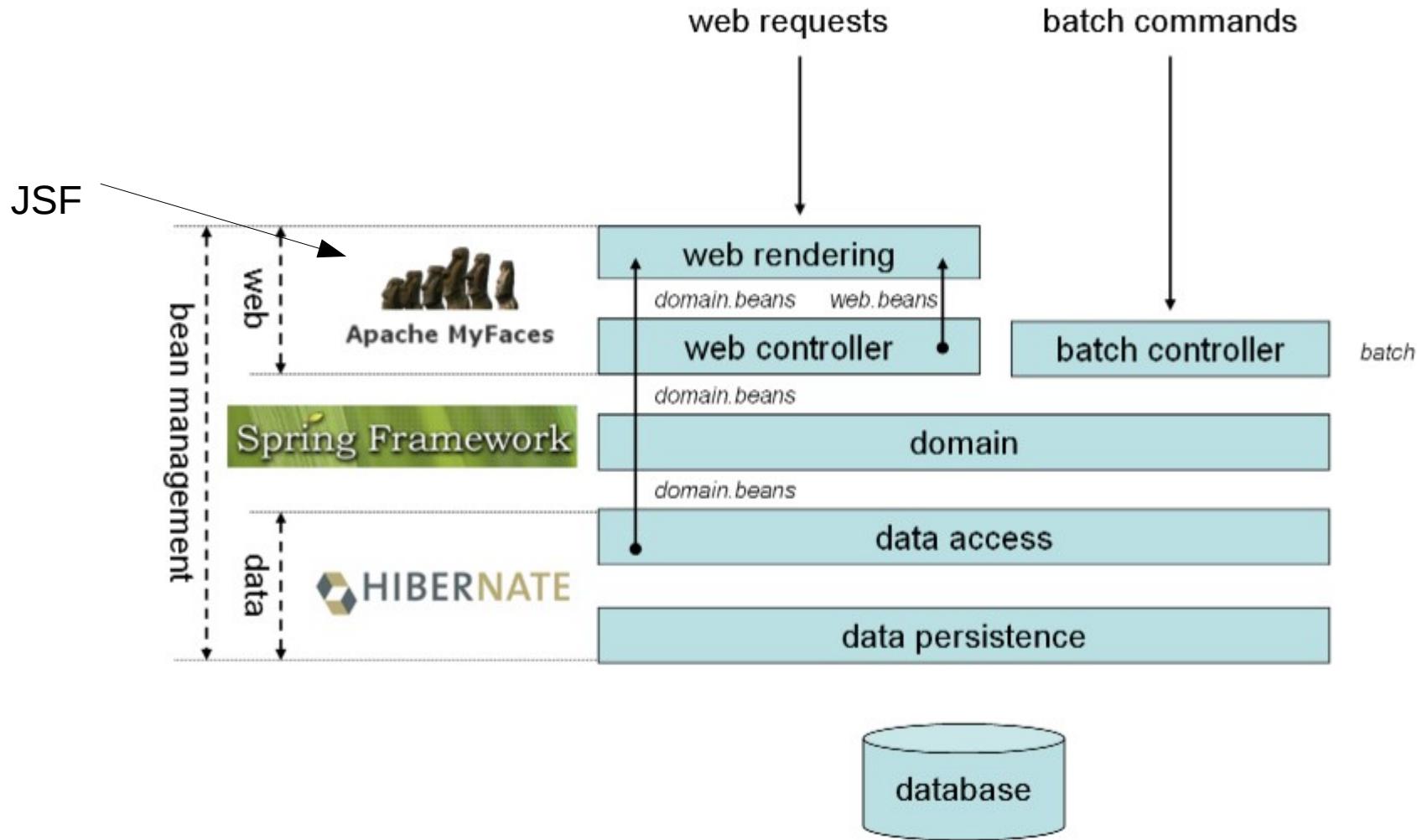
Programme

- Présentation du projet esup-commons
- Prise en main
- Organisation des fichiers
- Trucs & astuces
- Manipulation

Esup-commons

- Librairie développée et distribuée par le consortium **ESUP Portail**
- 100% java
 - SPRING pour la gestion des beans
 - JSF pour le rendu
 - Hibernate pour la gestion des bases de données

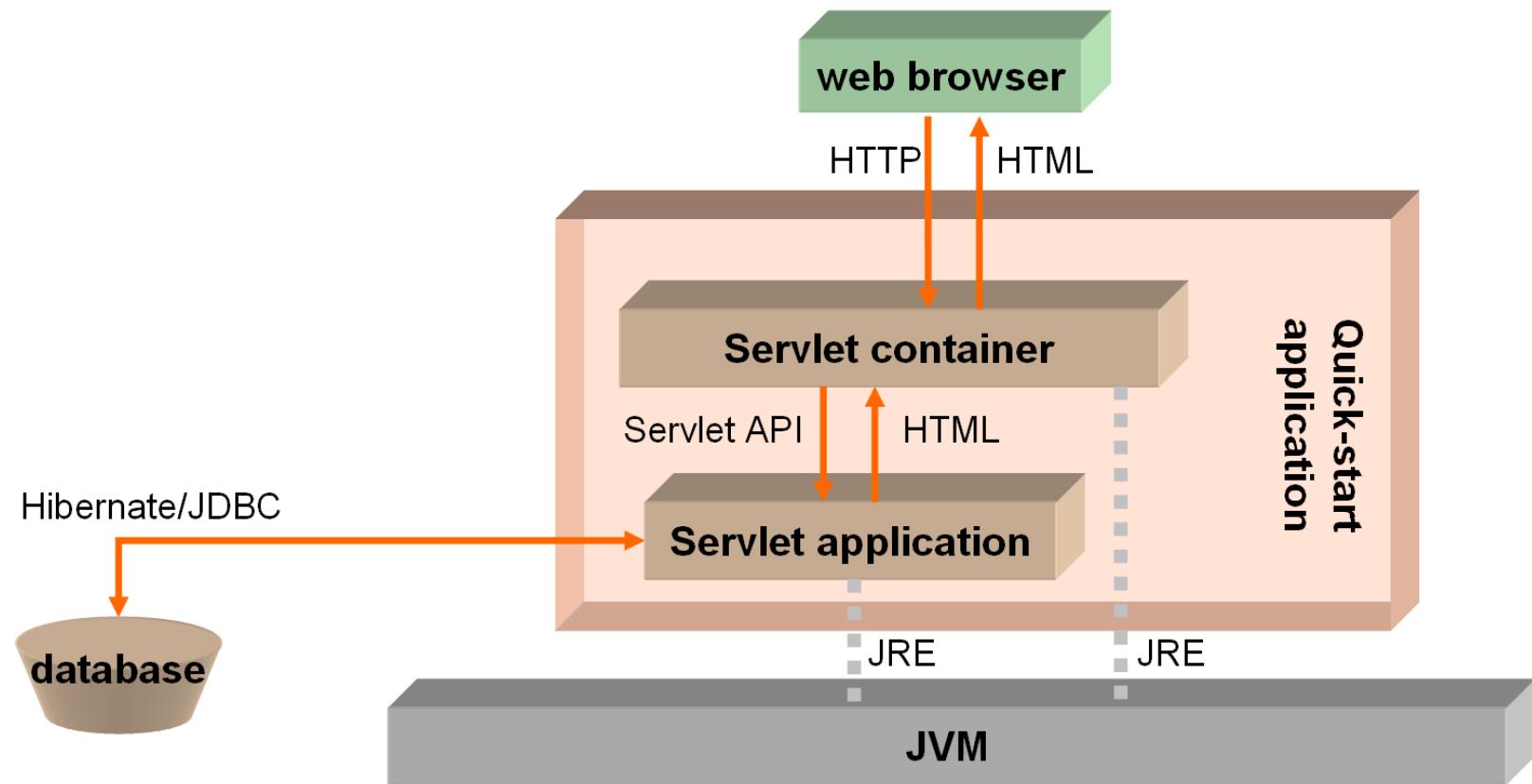
Architecture



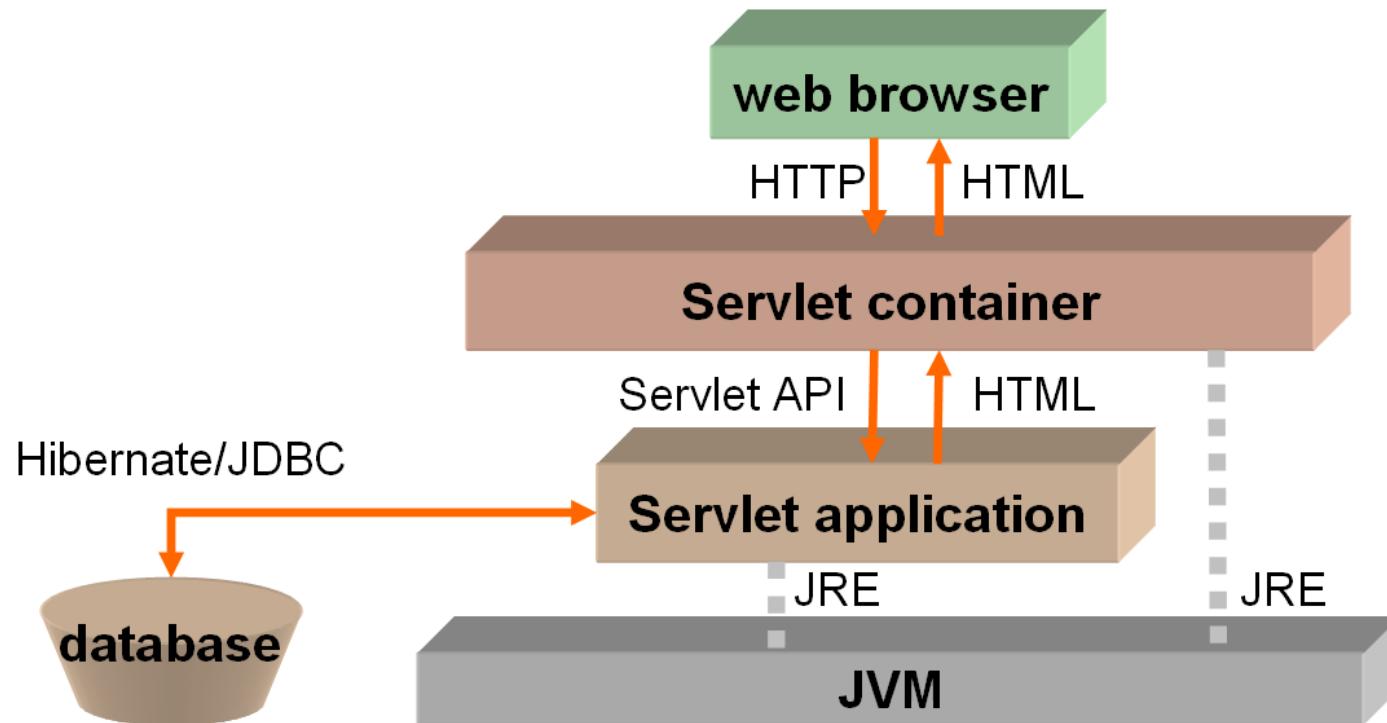
Déploiement

- Trois modes de déploiement sont disponibles:
 - Quick-start
 - Le serveur d'applications est déployé en même temps que l'application
 - L'application est indépendante du portail
 - Servlet
 - L'application est déployée sur un serveur d'applications existant
 - L'application est indépendante du portail
 - Portlet
 - L'application est déployée dans le même serveur d'applications que le portail

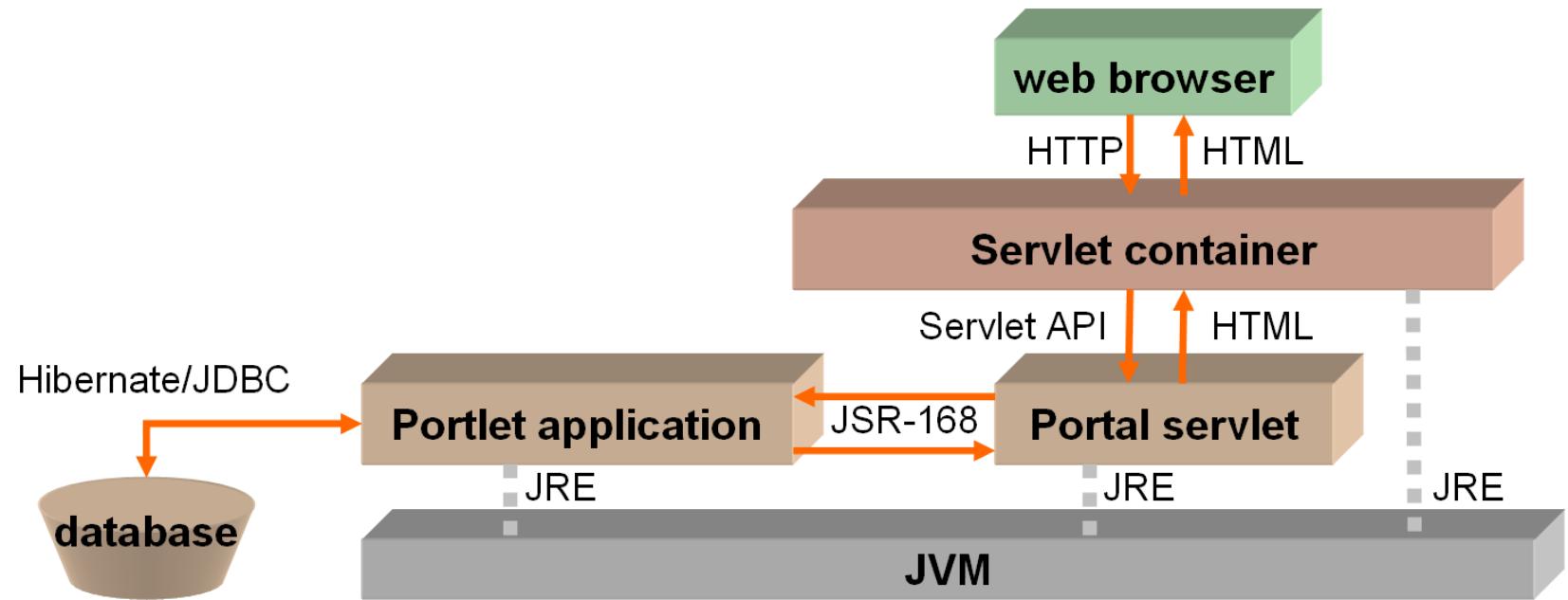
Déploiement Quick-Start



Déploiement Servlet



Déploiement Portlet



Méthodologie de développement

- <http://www.esup-portail.org/pages/viewpage.action?pageId=1867886>
- Le dépôt SVN du projet esup-commons est constitué de trois sous-projets :
 - La bibliothèque esup-commons,
 - L'application d'exemple esup-example,
 - L'application blanche esup-blank.
- Les trois projets sont récupérés depuis le svn
 - Esup-commons et Esup-example restent branchés sur le SVN.
- On utilise la base du projet esup-blank pour créer une nouvelle application.
 - Débranché du SVN
 - Renommé via la tâche *ant rename-application*
 - Éventuellement branché sur un nouveau dépôt SVN

Prise en main

- Eclipse
- ANT
- SVN
- SPRING
- HIBERNATE
- JSF

Eclipse

- <http://www.esup-portail.org/display/PROJCOMMONS/01+Installation+d%27Eclipse+%28Ganymede%29>
- <http://www.esup-portail.org/display/PROJCOMMONS/1.1.3+Installation+d%27Eclipse+%28Helios%29>
- Version installée sur la VM :
 - HELIOS
 - <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/heliossr2>
 - Compilateur Java configuré (menu Window > Preferences > Java > Compiler > Errors/Warnings)
 - Javadoc (menu Window > Preferences > Java > Compiler > Javadoc)
 - Editeur (complétion automatique sur les fichiers xhtml qui sont utilisés pour les facelets)
 - Préférences d'Eclipse
 - Ouvrir "général" --> "Content Types".
 - Ouvrir "Text"
 - Choisir "JSP"
 - Cliquer sur "Add"
 - Ajouter le type "*.xhtml".

Eclipse

- Modules complémentaires installés sur la VM
 - Spring IDE (core uniquement)
 - <http://www.esup-portail.org/display/PROJCOMMONS/1.1.3+Installation+d%27Eclipse+%28Helios%29#1.1.3Installationd%27Eclipse%28Helios%29-SpringIDE>
 - Checkstyle
 - <http://www.esup-portail.org/display/PROJCOMMONS/1.1.3+Installation+d%27Eclipse+%28Helios%29#1.1.3Installationd%27Eclipse%28Helios%29-Checkstyle>
 - Subversion
 - <http://www.esup-portail.org/display/PROJCOMMONS/1.1.3+Installation+d%27Eclipse+%28Helios%29#1.1.3Installationd%27Eclipse%28Helios%29-SubVersion>
- Projets installés (téléchargés depuis le dépôt SVN)
 - <http://www.esup-portail.org/pages/viewpage.action?pageId=1867898>
 - Esup-commons (version 1.3.9)
 - Esup-blank (Version 1.3.9)
 - Esup-example (Version 1.3.9)

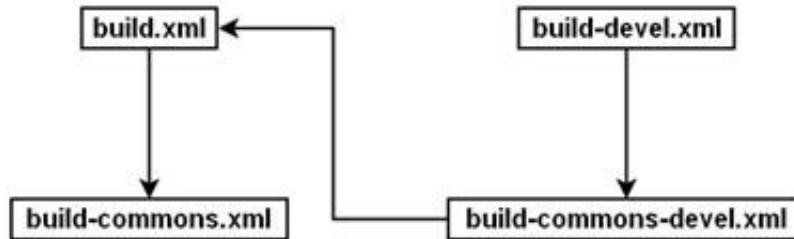
ANT

- Utilitaire java en remplacement de make
- Projet apache (<http://ant.apache.org>)
- Très riche en fonctionnalités
- Toutes plate formes (supportant java)
- Nécessite les variables d'environnement
 - JAVA_HOME
 - ANT_HOME
- Utilise un fichier de configuration XML
- Possibilité de fichiers de propriétés
- commande utile : ant -projecthelp

ANT - esup-commons

- <http://www.esup-portail.org/pages/viewpage.action?pageId=5210255>

- Les tâches ant en mode développement
 - build-devel.xml



- Les tâches du build.xml sont incluses dans le build-devel.xml
- Utiliser les tâches `_*`
- Les tâches ant en mode exploitation
 - build.xml

ANT – mode développement

- _clean
- _compile
- _deploy
- _dist
- _doc
- _prepare
- _rename-application
- _restart
- _start
- _stop
- _upload-doc

ANT – mode développement

- Tâche **_rename-application**

- Fichier *renameApplication.properties*

```
org/esupportail/blank=org/esupportail/myapp
org.esupportail.blank=org.esupportail.myapp
edu/domain/blank=edu/domain/myapp
ESUP-Portail\ Blank\ Application=ESUP-Portail My Application
project\ name\="esup-blank=project name="esup-myapp
logo_blank=logo_myapp
esup-blank=esup-myapp
```

- Rafraîchir le projet.

ATTENTION : La tâche **_rename-application** ne doit être lancée qu'une seule fois.

ANT – mode exploitation

check-version	check the database version
clean	clean the project
compile	compile the project
compile-batch	compile the project for a batch only
compile-web	compile the project for the web
deploy	deploy the application
esup-commons.test-beans	test all the beans
esup-commons.test-database	test the database connection
esup-commons.test-ldap	test the LDAP connection
esup-commons.test-portal	test the portal connection
esup-commons.test-smtp	test the SMTP connection
init-data	initialize the database
recover-config	recover configuration files from an older version
start	Start the application
stop	Stop the application
test-beans	test all the beans
test-config	test the configuration
test-config-batch	test the configuration for a batch only
test-database	test the database connection
test-ldap	test the LDAP connection
test-portal	test the portal connection
test-smtp	test the SMTP connection
upgrade	upgrade the database
Default target: compile	

SVN

- Système de gestion de version
 - [http://fr.wikipedia.org/wiki/Subversion_\(logiciel\)](http://fr.wikipedia.org/wiki/Subversion_(logiciel))
 - <http://subversion.tigris.org/>
- Licence APACHE et BSD
- Remplace CVS
- Permet de travailler en collaboration sur un projet
- Permet de garder les différentes versions d'un projet
- SVN ESUP-Portail hébergé au CRU
 - [https://sourcesup.cru.fr/projects/\[Nom_Projet\]](https://sourcesup.cru.fr/projects/[Nom_Projet])
 - [https://subversion.cru.fr/\[Nom_Projet\]](https://subversion.cru.fr/[Nom_Projet])

SPRING

- Framework Open Source
- Simplifier les développements
- Structurer les développements
- Programmation par aspects (POA)
 - http://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_aspect
- <http://www.springframework.org/node/155>
- <http://www.esup-portail.org/display/PROJCOMMONS/09+Les+beans+Spring>

SPRING

- fichier de configuration principal
 - */properties/applicationContext.xml*
 - déclaré dans le *web.xml*
- Déclaration des fichiers de configuration spécialisés dans le fichier principal
 - `<import resource="exceptionHandling/exceptionHandling.xml" />`
- Injection des données : Renseigner les attributs d'un bean via un fichier de configuration
 - Ne pas oublier d'ajouter les « *setter* » dans le bean pour accéder aux attributs.
 - Injection de chaînes de caractères
 - Injection de beans
 - Injection de listes
 - Héritage de configuration
- Portée des beans
 - Singleton : par défaut
 - Session : une instance par session utilisateur : Le plus utilisé pour les contrôleurs
 - Request : une instance par requête HTTP

SPRING

Exemple 1 : configuration du gestionnaire d'exceptions:

```
<bean
  id="exceptionServiceFactory"
  class="org.esupportail.formation.services.exceptionHandling.CachingEmailExceptionServiceFactoryImpl"
  parent="abstractApplicationAwareBean">
  <property
    name="doNotSendExceptionReportsToDevelopers"
    value="false"/>
  <property name="smtpService" ref="smtpService"/>
  <property name="authenticationService" ref="authenticationService"/>
  <property name="recipientEmail" value="webmaster@domain.edu"/>
  <property name="cacheManager" ref="cacheManager"/>
  <property name="cacheName" value="" />
</bean>
```



Exemple 2 : Injection d'une liste :

```
<property name="servers">
  <list>
    <ref bean="smtpServer1" />
    <ref bean="smtpServer2" />
  </list>
</property>
```

Exemple 3 : Héritage de configuration :

```
<bean
  id="abstractApplicationAwareBean"
  parent="abstractI18nAwareBean"
  abstract="true">
  <property name="applicationService" ref="applicationService" />
</bean>
```

Hibernate

- Framework Open Source de gestion de base de données relationnelle
- Gestion de la persistance des objets en base de données
- Gestion des requêtes sur la base de données
- Gestion du lien table/bean
 - tables : table de la base de données
 - beans : objet java représentant la table et ses colonnes

Hibernate Mapping

auteur

```
id_auteur integer
nom character varying(255)
prenom character varying(255)
```

titre

```
id_titre integer
valeur character varying(255)
isbn varying(255)
id_auteur integer
```

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="org.esupportail.livres.domain.beans">
  <class name="Auteur" table="auteur">
    <id name="idAuteur" column="id_auteur" type="integer">
      <generator class="increment"/>
    </id>
    <property name="nom" column="nom" type="string"/>
    <property name="prenom" column="prenom" type="string"/>
    <set name="livres" table="livre" inverse="true" lazy="true" fetch="select">
      <key>
        <column name="id_auteur" not-null="true" />
      </key>
      <one-to-many class="Livre" />
    </set>
  </class>
</hibernate-mapping>
```

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="org.esupportail.livres.domain.beans">
  <class name="Titre" table="titre">
    <id name="idTitre" column="id_titre" type="integer">
      <generator class="increment"/>
    </id>
    <property name="valeur" column="valeur" type="string"/>
    <property name="isbn" column="isbn" type="string"/>
    <many-to-one name="idAuteur" class="Auteur" column="id_auteur"/>
  </class>
</hibernate-mapping>
```

Hibernate Bean

ATTENTION:

- nomVariable : commence **toujours** par une **minuscule**
- Nom de méthode getNomVariable et setNomVariable

```
public class Auteur {  
    private int idAuteur;  
    private String nom;  
    private String prenom;  
    private Set livres = new HashSet(0);  
  
    /* les methodes get et set */  
    public int getIdAuteur(){...}  
    public void setIdAuteur(int idAuteur){...}  
    public String getNom(){...}  
    public void setNom(String nom){...}  
    public String getPrenom(){...}  
    public void setPrenom(String prenom){...}  
    public Set getLivres() {...}  
    public void setLivres(Set livres) {...}  
}
```

```
public class Titre {  
    private int idTitre;  
    private String valeur;  
    private String isbn;  
    private int idAuteur;  
  
    /* les methodes get et set */  
    public int getIdTitre(){...}  
    public void setIdTitre(int idTitre){...}  
    public String getValeur(){...}  
    public void setValeur(String valeur){...}  
    public String getIsbn(){...}  
    public void setIsbn(String isbn){...}  
    public int getIdAuteur(){...}  
    public void setIdAuteur(int idAuteur){...}  
}
```

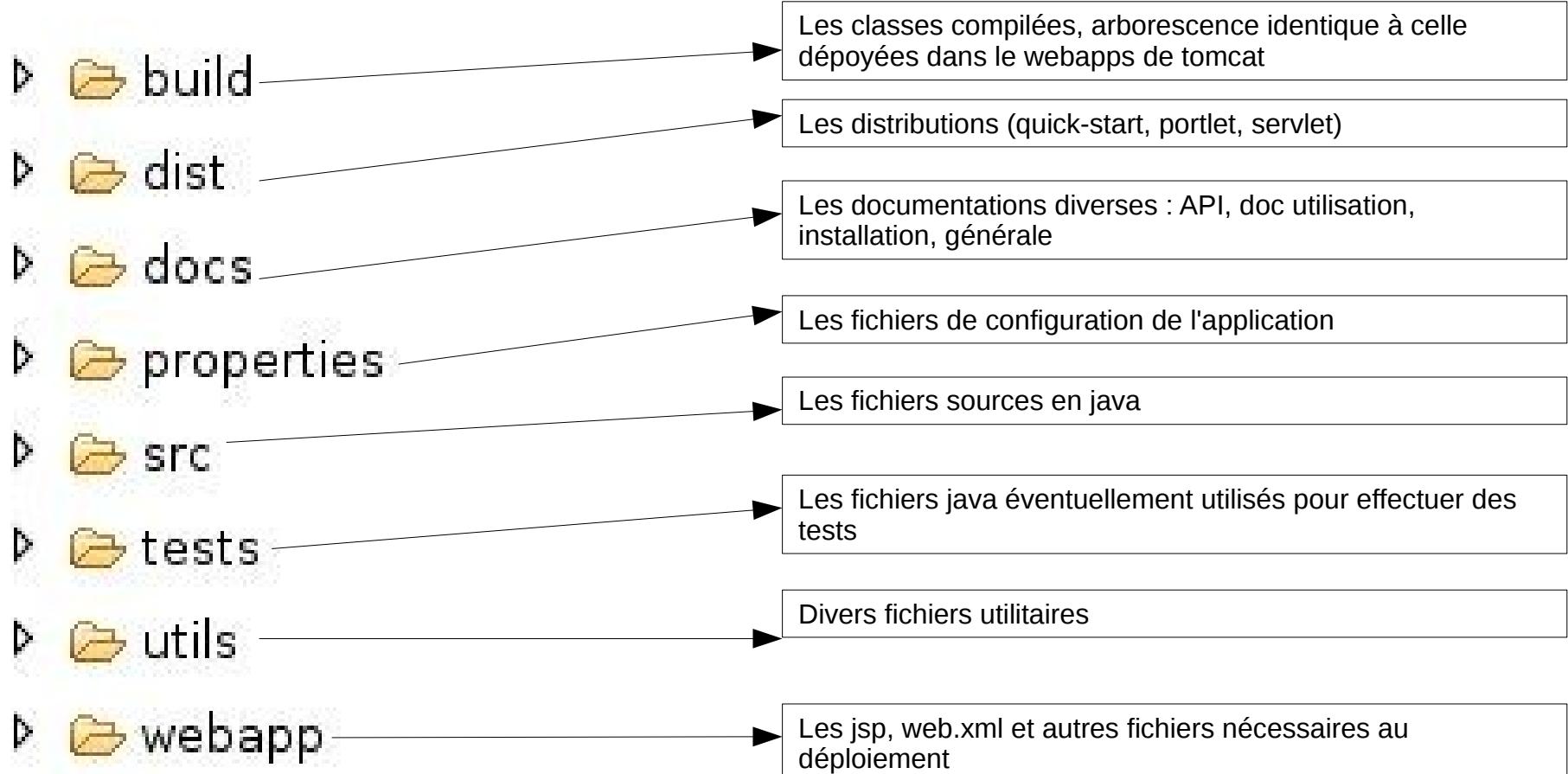
JSF

- Séparation entre les couches présentation et métier
- Lien direct entre le rendu HTML et les objets gérés
- basé sur des pages JSP
- Des librairies (tagLib) disponibles:
 - librairie H
 - librairie T
 - librairie e -> librairies esup-commons basée sur h
- <http://www.esup-portail.org/pages/viewpage.action?pageId=6684706>
- Navigation entre les pages
 - */properties/jsf/navigation-rules.xml*

Organisation des fichiers

- Fichiers de configuration
- Fichiers sources
- Gestion de l'IHM
 - JSF
 - Media
 - CSS

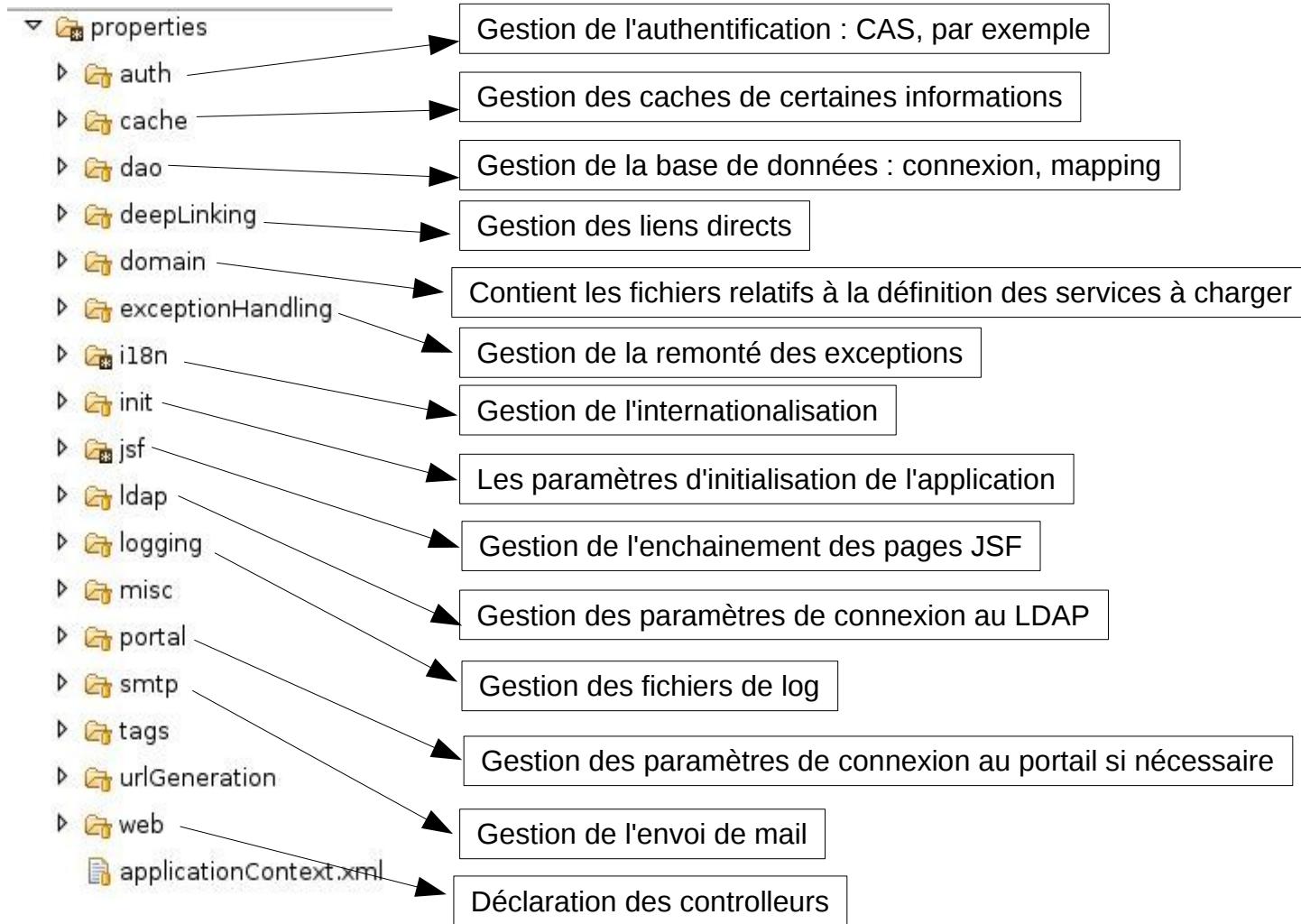
Organisation des fichiers



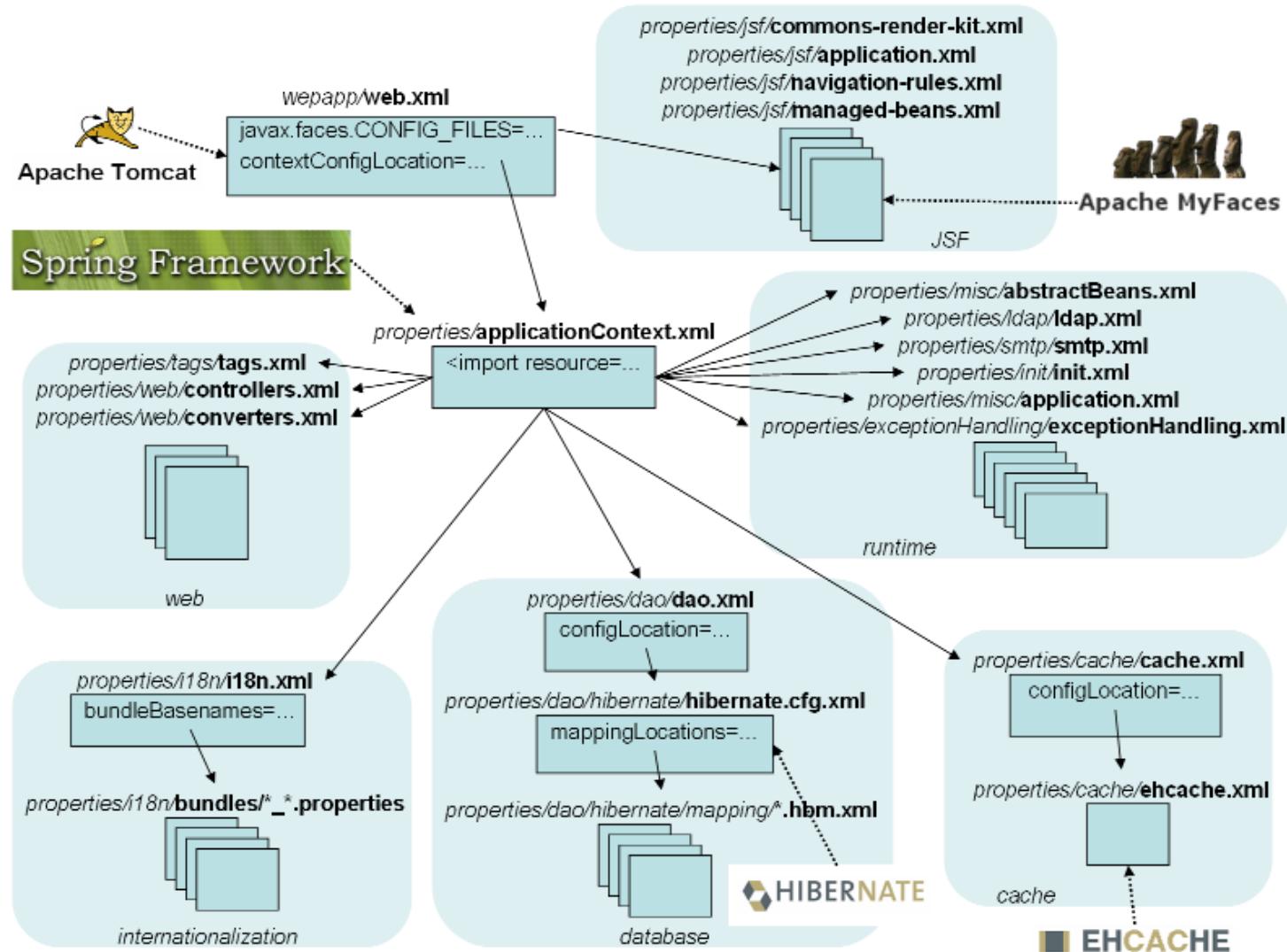
Fichiers de configuration

- <http://www.esup-portail.org/display/PROJCOMMONS/08+Organisation+des+fichiers#08Organisationdesfichiers-%2Fproperties%3Alesfichiersdeconfiguration>
- Les fichiers de configuration sont en général fournis sous forme de fichiers d'exemple
 - xxx-example.xml
 - En mode développement, ces fichiers d'exemple se trouvent
 - soit dans la hiérarchie du projet esup-commons (s'ils ne sont pas particulier à une application)
 - soit dans le projet de votre application (si la configuration par défaut est particulière).
- La plupart des informations à modifier le sont dans le fichier de configuration principal
 - config.properties

Fichiers de configuration



Fichiers de configuration



L'authentification properties/auth/

- Les informations modifiables pour la configuration de l'accès au SSO sont données dans le *config.properties*

```
auth.portal.uidAttribute=uid  
cas.url=https://cas.univ-xxxxx.fr/cas
```

L'authentification properties/auth/auth.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd"
    >
    <bean id="authenticator" lazy-init="true"
        class="org.esupportail.cil.services.authentication.AuthenticatorImpl">
        <property name="authenticationService" ref="${auth.bean}" />
        <property name="domainService" ref="domainService" />
    </bean>

    <bean id="servletAuthenticationService" lazy-init="true"
        class="org.esupportail.common.services.authentication.CasFilterAuthenticationService">
    </bean>

    <bean id="portletAuthenticationService" lazy-init="true"
        class="org.esupportail.common.services.authentication.CasifiedPortalAuthenticationService">
        <property name="uidPortalAttribute" value="${auth.portal.uidAttribute}" />
    </bean>
</beans>
```

- La configuration pour les développements ne nécessitant pas d'authentification via le SSO est disponible dans les fichiers *d'exemples*

Le cache /properties/cache

- cache.xml
 - détermine la classe utilisée pour gérer les caches
 - détermine le fichier de configuration des données à mettre en cache
 - n'est pas nécessaire de modifier
- ehcache.xml
 - Détermine les paramètres pour chaque bean qui doit être mis en cache
 - Exemple:

```
<cache
  name="org.esupportail.cnil.domain.beans.User"
  maxElementsInMemory="1000"
  eternal="false"
  timeToIdleSeconds="30"
  timeToLiveSeconds="60"
  overflowToDisk="true"
/>
```

Les fichiers de configuration properties/dao/

- Les informations modifiables pour la configuration de l'accès à la base de données sont dans le *config.properties*

```
hibernate.connection.driver_class=org.postgresql.Driver
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.connection.jdbc.url=jdbc:postgresql://localhost:5432/biblio
hibernate.connection.jdbc.username=xxxxx
hibernate.connection.jdbc.password=xxxx
hibernate.jdbcTemplateBeanName=jdbcHibernateTemplate
hibernate.jdbcSessionFactoryBeanName=jdbcSessionFactory
hibernate.jdbcUpgradeSessionFactoryBeanName=jdbcUpgradeSessionFactory
hibernate.useJndi=true
hibernate.connection.jndi.datasource=jdbc/esup-biblio
hibernate.jndiTemplateBeanName=jndiHibernateTemplate
hibernate.jndiSessionFactoryBeanName=jndiSessionFactory
hibernate.jndiUpgradeSessionFactoryBeanName=jndiUpgradeSessionFactory
```

- Des exemples de configuration pour les différents types de SGBD sont disponibles dans les fichiers *d'exemples*

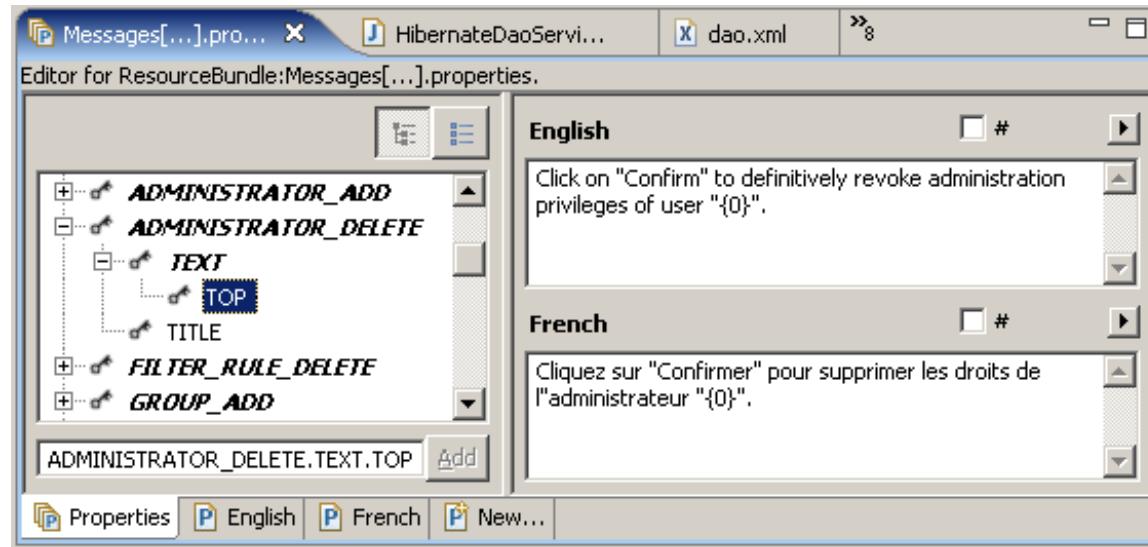
Les fichiers de configuration properties/dao/dao.xml

Déclaration des fichiers mapping traités par hibernate

```
<bean
    id="abstractHibernateSessionFactory"
    abstract="true"
    class="org.springframework.orm.hibernate3.LocalSessionFactoryBean"
    >
    <description>
        This bean is common to all the session factories (see below). It allows
        a single declaration of the mapping files.
    </description>
    <property name="configLocation"
        value="classpath:/properties/dao/hibernate/hibernate.cfg.xml" />
    <property name="mappingLocations">
        <list>
            <value>
                classpath:/properties/dao/hibernate/mapping/User.hbm.xml
            </value>
            ....
        </list>
    </property>
</bean>
```

Les fichiers de configuration properties/i18n

- Fichier de configuration de l'internationalisation
- i18n/bundles/
 - Custom_en.properties
 - Custom_fr.properties
 - Messages_en.properties
 - Messages_fr.properties
- Utilise le plugin eclipse : Ressource Bundle



le MVC et la présentation /properties/jsf

- Le fichier navigation-rules.xml définit les règles de navigation entre les pages de l'application.

```
<!-- Gestion des auteurs -->
<navigation-rule>
  <display-name>listeAuteur</display-name>
  <from-view-id>/stylesheets/auteur/listeAuteurs.jsp</from-view-id>

  <navigation-case>
    <from-outcome>detailAuteur</from-outcome>
    <to-view-id>/stylesheets/auteur/detailAuteur.jsp</to-view-id>
  </navigation-case>

  <navigation-case>
    <from-outcome>modifAuteur</from-outcome>
    <to-view-id>/stylesheets/auteur/modifAuteurs.jsp</to-view-id>
  </navigation-case>

  <navigation-case>
    <from-outcome>addAuteur</from-outcome>
    <to-view-id>/stylesheets/auteur/addAuteurs.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<!-- Gestion des titres -->
<navigation-rule>
  <display-name>addTitre</display-name>
  <from-view-id>/stylesheets/titre/addAuteurs.jsp</from-view-id>

  <navigation-case>
    <from-outcome>addAuteur</from-outcome>
    <to-view-id>/stylesheets/auteur/addAuteurs.jsp</to-view-id>
  </navigation-case>
...
</navigation-rule>
```

La page **addAuteur** est accessible depuis les pages **listeAuteur** et **addTitre**

Les fichiers de configuration properties/LDAP

- Les informations modifiables pour la configuration du LDAP sont données dans le *config.properties*

```
ldap.url=ldap://ldap.univ-xxxxxx.fr:389
ldap.userName=
ldap.password=
ldap.base=dc=univ-xxxxx,dc=fr
ldap.dnSubPath=ou=people
ldap.uidAttribute=uid
ldap.displayNameAttribute=displayName
ldap.emailAttribute=mail
ldap.searchAttribute=cn
ldap.testFilter=cn=*_uidtest*
ldap.connectTimeout=5000
ldap.attributes=cn,displayName,supannAffectation,postalAddress,mail,PersCategorie,EtuComposante,EtuFormation,telephoneNumber
ldap.searchDisplayedAttributes=cn,mail,supannAffectation
```

Les fichiers de configuration properties/logging

- Un fichier de log par application
- Gestion des logs en mode batch et en mode n/tiers
 - Mode batch : *log4j-batch.properties*
 - Par défaut sur la sortie standard.
 - Mode servlet/portlet : *log4j.properties*
 - Informations récupérées depuis le *config.properties*

```
log.level=WARN
log.output=stdout
log.pattern=%d %p [%c] - %m%n
log.file=esup-blank.log
log.maxFileSize=5MB
log.maxBackupIndex=3
```

Les fichiers de configuration properties/portal

- Permet d'accéder aux données remontées par le WS
 - esup-portal-ws
- Accès aux groupes et attributs utilisateurs du portail
- Définir l'accès aux données du portail
 - Pas défini par défaut dans esup-blank
 - <http://www.esup-portail.org/pages/viewpage.action?pageId=6684863>
- Un exemple de configuration est disponible dans les fichiers *d'exemples*

Les fichiers de configuration properties/smtp

```
smtp.host=smtp.domain.edu
smtp.port=25
smtp.user=
smtp.password=
smtp.charset=utf-8
smtp.fromEmail=devel-test@univ-valenciennes.fr
smtp.fromName=Portlet test
smtp.interceptAll=true
smtp.interceptEmail=doriane.dusart@univ-valenciennes.fr
smtp.interceptName=Dusart Doriane
smtp.notInterceptedAddresses=
```

From: Portlet test <devel-test@univ-valenciennes.fr>
To: "Dusart Doriane (normally sent to portail@univ-valenciennes.fr)" <doriane.dusart@univ-valenciennes.fr>

Définie dans le fichier
de gestion des exceptions
Via le config.properties

```
exceptionHandling.logLevel=INFO
exceptionHandling.email=portail@univ-valenciennes.fr
```

Les fichiers de configuration properties/web/

- controller.xml

- Définition des contrôleurs

```
<bean id="NOM_CONTROLLER"
      class="org.esupportail.cnil.web.controllers.NOM_CONTROLLER"
      parent="abstractContextAwareController" ←
      scope="session"
    >
    <description>A bean to Add a Declaration.</description>
</bean>
```

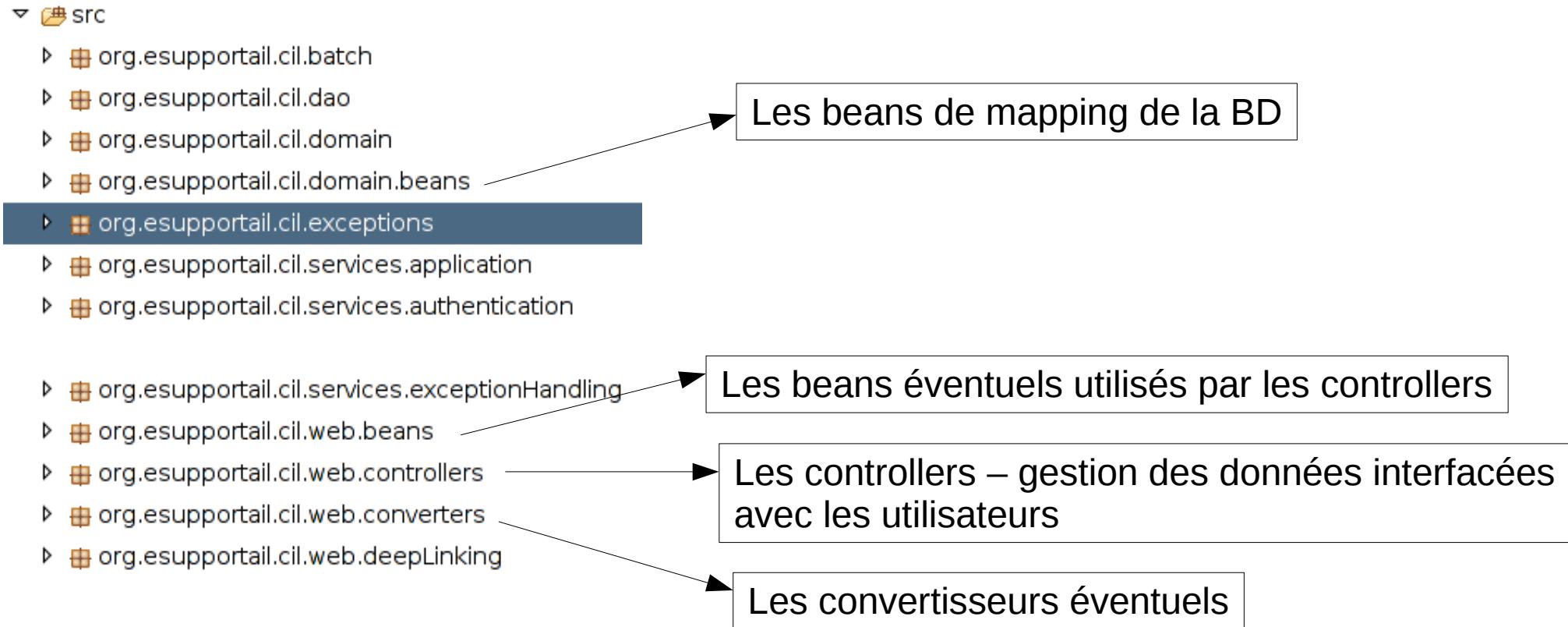
Peut être un autre contrôleur dont héritera celui-ci

- converters.xml

- Définition des convertisseurs de données. Utilisés par exemple pour les listes de choix dans les formulaires lorsque l'on doit manipuler des objets complexes. Un exemple d'utilisation est disponible dans le fichier `exemples_configurations.pdf`

```
<bean
      id="localeConverter"
      class="org.esupportail.commons.web.converters.LocaleConverter"
    >
    <description>
      A converter for Locale objects.
    </description>
</bean>
```

Les fichiers sources



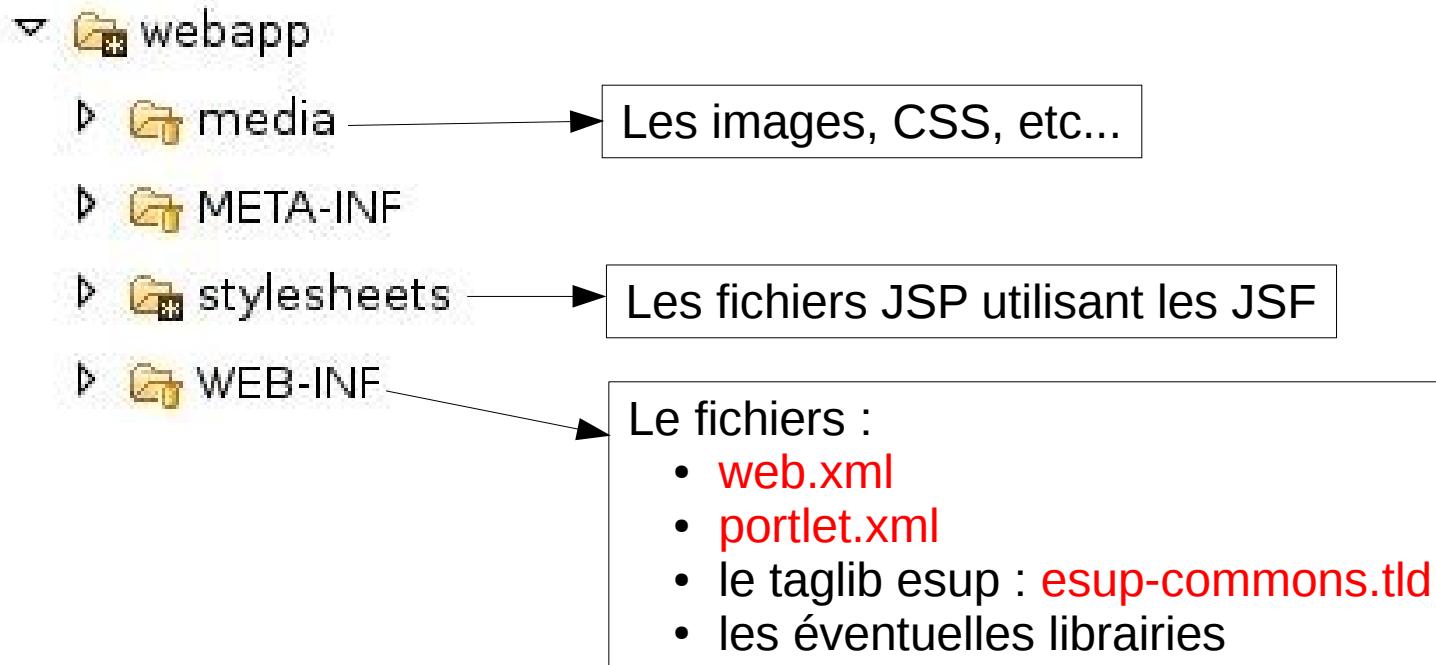
Les beans de mapping

- Les beans de mapping utilisés par hibernate
- Par défaut
 - User.java
 - VersionManager.java

Les controllers

- Gestion des données transmises ou récupérées des IHM
- Récupération des données de la BD via les beans
- Récupération des données LDAP
- ...

Les fichiers d'IHM



Les JSF

- <http://www.esup-portail.org/display/PROJCOMMONS/10+JSF+%28Java+Server+Faces%29>
 - Fichier *esup-commons.tld*
 - Librairie h pour référence complète
 - Utilisation des « facelets »
 - Librairie tomahawk
 - Les principales utilisations
 - Les formulaires
 - Gestion simplifiée des tableaux
 - Gestion simplifiée des menus

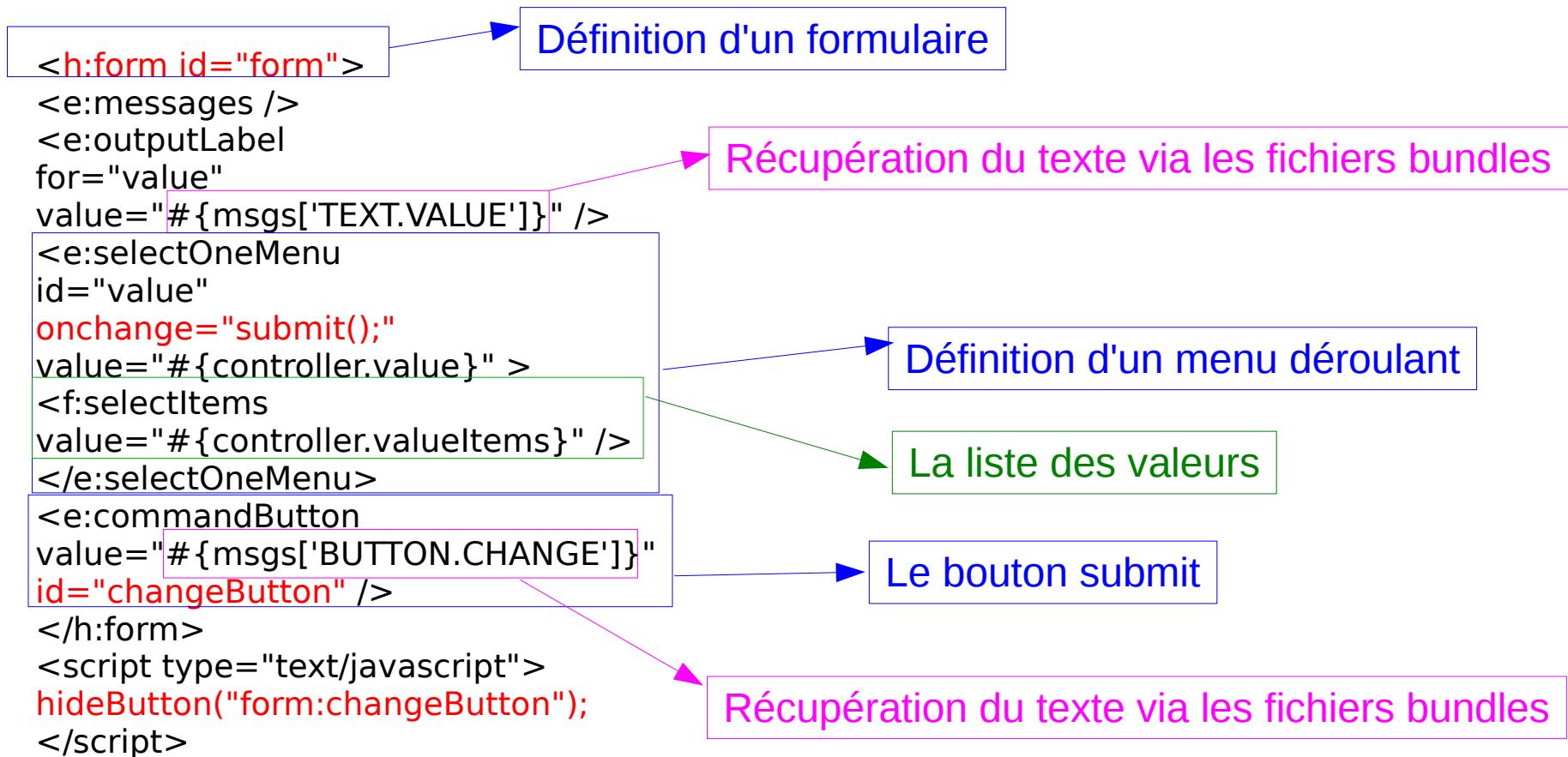
Exemple

```
<e:page  
locale="#{sessionController.locale}"  
stringVars="msgs"  
menuItem="welcome" >
```

CONTENU DE LA PAGE

```
</e:page>
```

Exemple Formulaire simple



Exemple Recherche LDAP

```
<e:inputText id="loginResponsable" value="#{AuteurController.uid}" />
```

```
<e:commandButton  
    value="#{msgs['_.BUTTON.LDAP']}"  
    action="ldapSearch"  
    immediate="true">
```

Nom de la variable
d'affectation

Nom du controller

```
<t:updateActionListener  
    value="#{AuteurController}"  
    property="#{ldapSearchController.caller}" />
```

```
<t:updateActionListener  
    value="userSelectedToAdd"  
    property="#{ldapSearchController.successResult}" />
```

La recherche LDAP

```
<t:updateActionListener  
    value="cancelToAdd"  
    property="#{ldapSearchController.cancelResult}" />
```

```
</e:commandButton>
```

Les vues qui
seront affichées
le cas échéant

Exemple

Afficher un tableau

```
<h:form id="projettaite">
<e: dataTable
    id="traitements"
    rowIndexVar="variable"
    value="#{TitresController.listeComplete}"
    var="TitresController"
    border="1"
    style="width:100%"
    cellspacing="0"
    cellpadding="0"
    >
<t:column
    <f:facet name="header">
        <e:text value="Titre"/>
    </f:facet>
    <e:text value="#{TitresController.titre.valeur}" />
</t:column>
<t:column>
    <f:facet name="header">
        <e:text value="ISBN" />
    </f:facet>
    <e:text value="#{TitresController.titre.isbn}" />
</t:column>
</e: dataTable>
```

Le contrôleur qui gère les données

La variable contenant les données à afficher

Le bean de définition des données

Les variables à afficher

Trucs & astuces

- Divers exemples
 - De configuration
 - D'utilisation des convertisseurs
 - D'utilisation des Paginateurs
 - ...
- CF : *les fichiers d'exemples*

Manipulation

- Configuration
 - **Esup-commons**
 - <http://www.esup-portail.org/pages/viewpage.action?pageId=1867898#04Cr%C3%A9ationduprojetsup-commons-Cr%C3%A9ationdespremiersr%C3%A9pertoires>
 - **Esup-blank**
 - <http://www.esup-portail.org/pages/viewpage.action?pageId=1867908#05Cr%C3%A9ationduprojetsup-blank-Configurationduprojet>
- Cration d'une application
- CF : exercice.pdf