|  |
| --- |
| Formation JSF par la pratique  Partie 4 : Intégration de Spring (implémentation de l'IOC et l'AOP) |
| Développement d'une application web avec Spring, JSF, JPA et WildFly |

Sommaire

[1. Objet du TP 3](#_Toc2956817)

[2. Pré requis 3](#_Toc2956818)

[3. Partie 1 : Préparation de l'environnement de développement 3](#_Toc2956819)

[4. Partie 2 : Développement de l'application web 4](#_Toc2956820)

[**Exercices :** 73](#_Toc2956821)

# Objet du TP

Le présent TP vous montre comment développer une application web avec Spring, JSF, Primefaces et JPA sous wildfly.

Nous allons voir comment implémenter :

* l'IOC (Inversion Of Control) pour que le couplage soit faible entre les classes.
* l'AOP (Aspect Oriented Programming) pour séparer les traitements techniques de la couche métier. Dans ce TP, nous allons séparer le traitement technique concernant la gestion des tranasactions de la couche métier.

# Pré requis

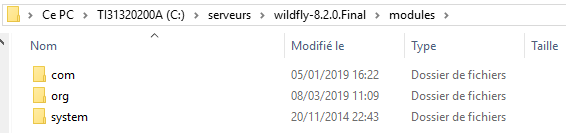
|  |  |
| --- | --- |
| Outil | Chemin du fichier  voir dossier : *formation\_Framework* |
| Eclipse Luna | formation\_Framework\tools\eclipse-luna.rar |
| WildFly 8.2.0 | formation\_Framework\tools\wildfly-8.2.0.Final.zip |
| Le serveur MySQL 5.0 | formation\_Framework\tools\MySQL 5.0\Setup.exe |
| Le client MySQL (l'interface graphique) | formation\_Framework\tools\MySQL 5.0\mysql-gui-tools-5.0-r17-win32.msi |
| JDK 1.7 | formation\_Framework\tools\jdk-7u75-windows-x64.exe |
| Le Driver MySQL 5.0 | formation\_Framework\tools\divers\mysql-connector-java-5.1.17-bin.jar |
| L'implémentation Hibernate de JPA. | formation\_Framework\tools\lib-Hibernate\\*.jar |
| - PrimeFaces  - Les thèmes  - Apache POI  - xmlbeans  - itext | formation\_Framework\tools\lib-Primefaces-Themes\\*.jar |
| Apache commons-io | formation\_Framework\tools\divers\commons-io-2.4.jar |
| Spring | formation\_Framework\tools\lib-Spring\\*.jar |
| Les photos utilisées dans ce TP. | formation\_Framework\TPs\images |

# Partie 1 : Préparation de l'environnement de développement

\* Idem que le TP n°1.

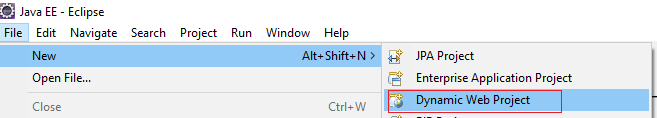
\* Copier le contenu du dossier modules (ce dernier se trouve dans *formation\_Framework\tools\modules*) dans le dossier *\wildfly-8.2.0.Final\modules* de votre serveur Wildfly.

Le contenu du dossier *wildfly-8.2.0.Final\modules* doit être :

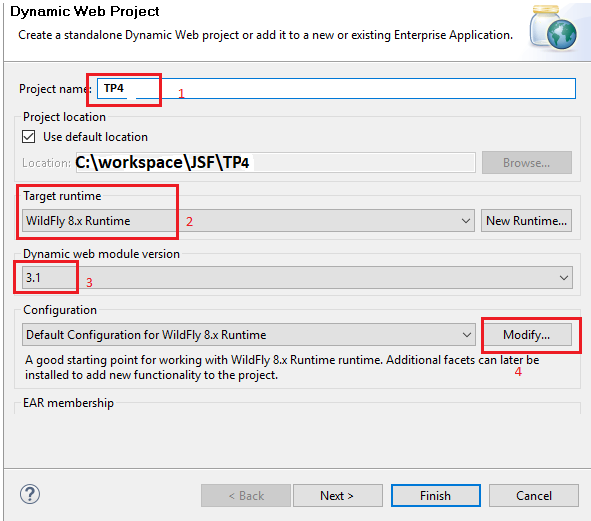


# Partie 2 : Développement de l'application web

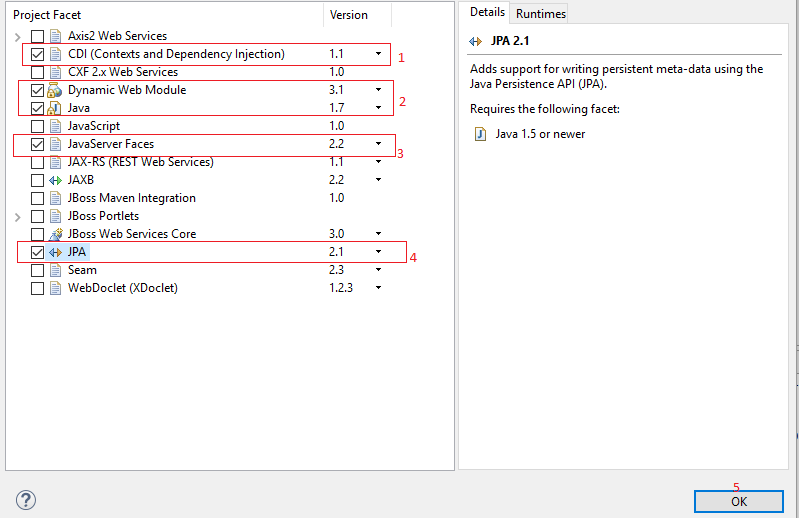
Au niveau d'Eclipse, créer un projet web comme illustre la fenêtre suivante :



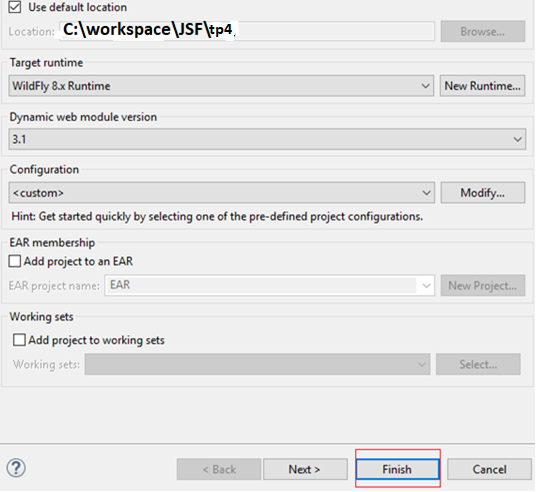
Cliquer sur "Dynamic Web Project" :



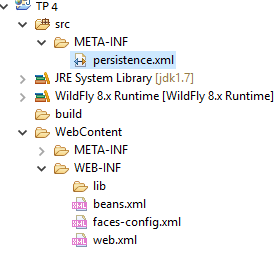
1. Saisir le nom de votre projet (ici : TP4);
2. Préciser la "Target runtime" Wildfly 8.x Runtime;
3. Préciser la version de la Servlet (ici : 3.1) ;
4. Cliquer ensuite sur Modifiy.... L'écran suivant sera affiché :



1. Cocher CDI (Context and Dependency Injection.
2. Vérifier que la version de "Dynamic Web Module" est bien 3.1 et la version de java est bien 1.7 ;
3. Cocher JavaServerFaces et préciser la version 2.2 ;
4. Cocher JPA et préciser la version 2.1 ;
5. Enfin, cliquer sur le bouton OK.

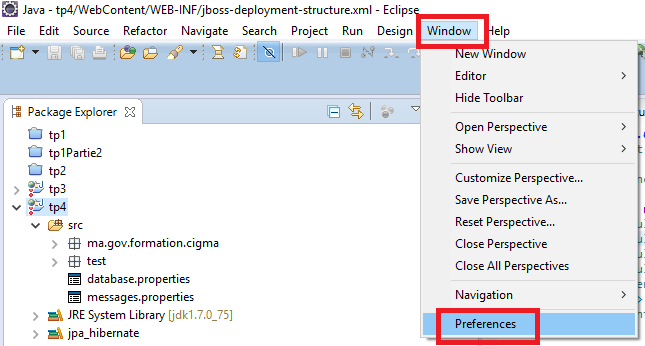


- Cliquer sur Finish. L'arborescence du projet ci-dessous est créée :

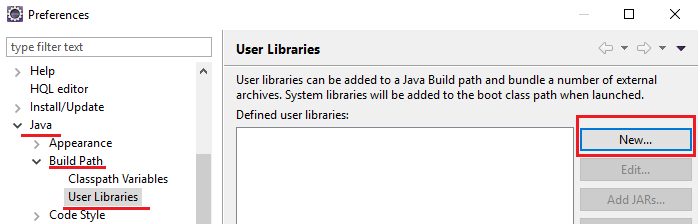


Créer les "User Libraries" suivantes : Divers, Primefaces, Spring et Hibernate en suivant les étapes suivantes :

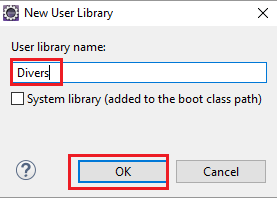
- Cliquer sur le menu Windows comme illustre l'écran suivant :



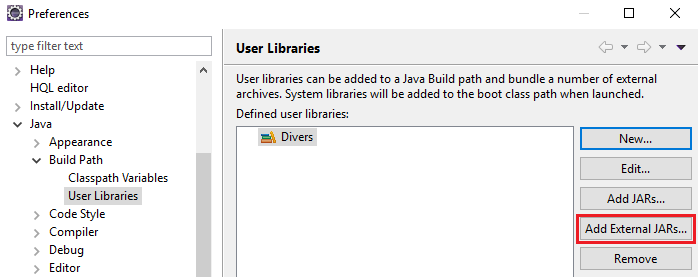
- Cliquer sur Preferences. L'écran suivant s'ouvre :



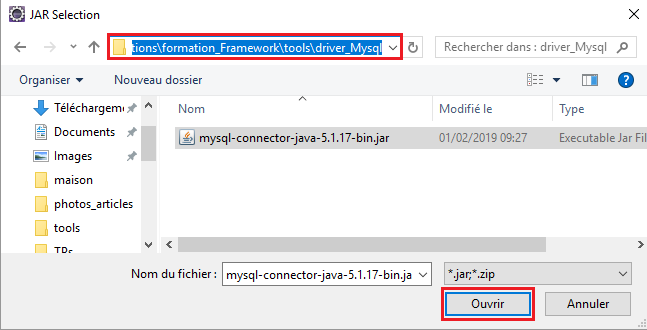
- Pour ajouter la bibliothèque utilisateur Divers, cliquer sur le bouton New... :



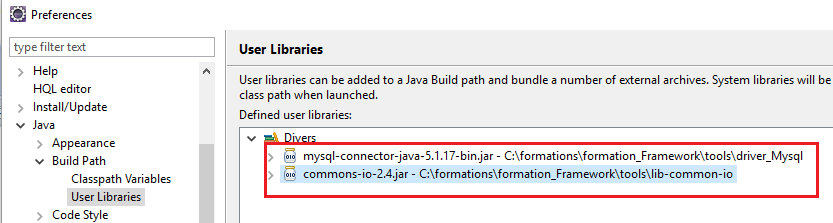
- Saisir le nom de votre bibliothèque (ici c'est Divers) et cliquer ensuite sur le bouton OK :



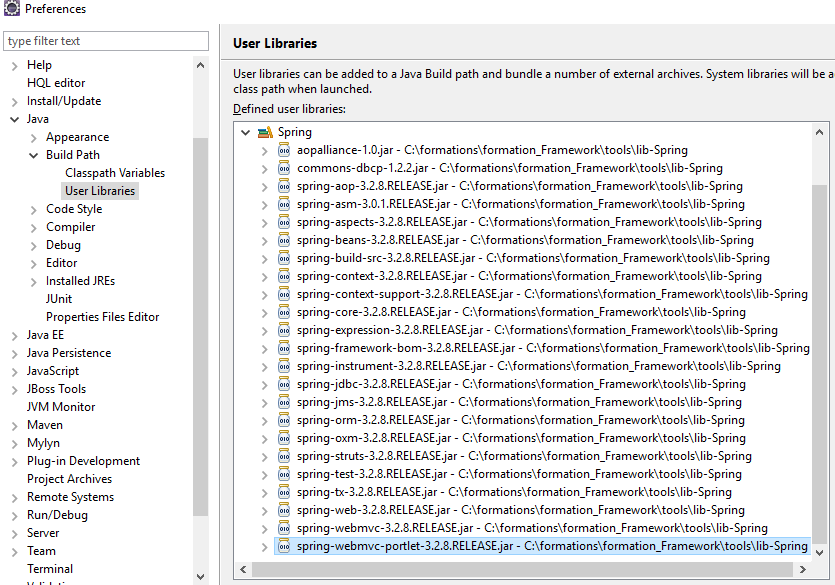
- Cliquer sur le bouton "Add External JARs..." :



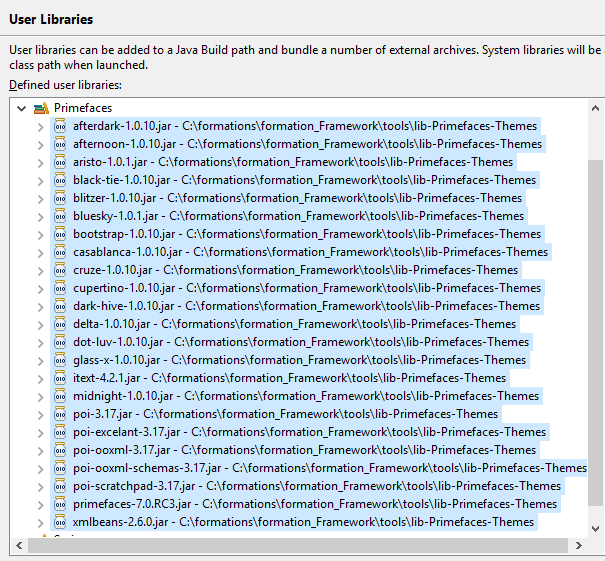
- Ajouter le driver de Mysql (formation\_Framework\tools\driver\_Mysql) et cliquer en suite sur Ouvrir. Refaire la même étape pour le JAR common-io. Le contenu de la bibiothèque utilisateur Divers doit être :



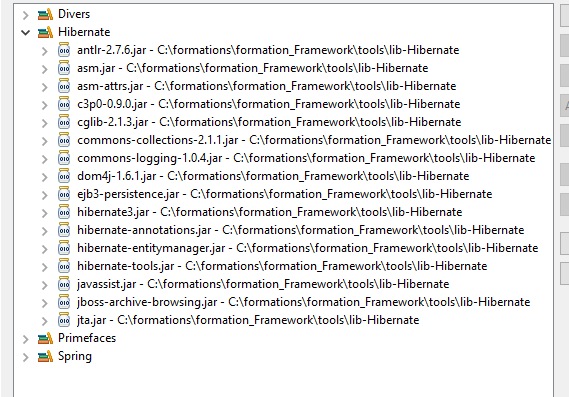
- De la même façon, créer la bibiothèque Spring. les JARs de Spring se trouvent dans le chemin formation\_Framework\tools\lib-Spring. Le contenu de la bibliothèque Spring doit être :



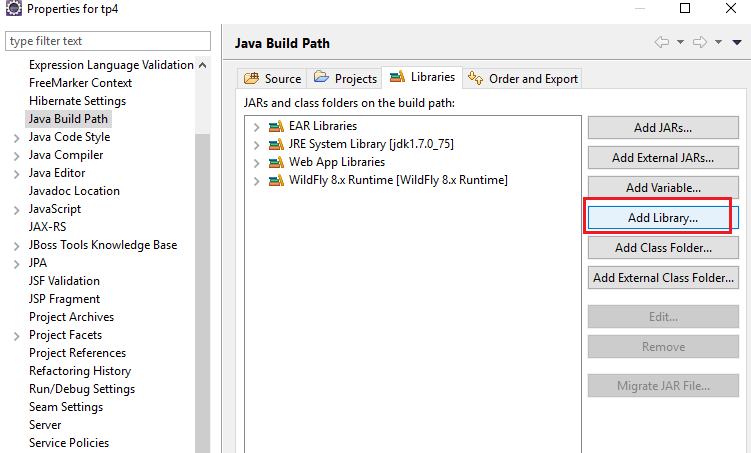
- De la même façon, créer la bibiothèque Primefaces. les JARs de Primefaces avec les thèmes utilisés se trouvent dans le chemin formation\_Framework\tools\lib-Primefaces-Themes. Le contenu de la bibliothèque Primefaces doit être :



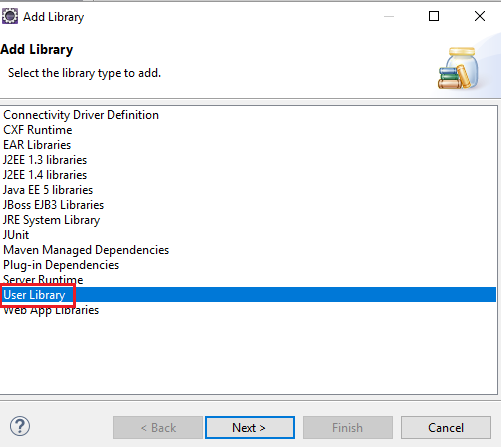
- De la même façon, créer la bibiothèque Hibernate. les JARs de Hibernate utilisés se trouvent dans le chemin formation\_Framework\tools\lib-Hibernate. Le contenu de la bibliothèque Hibernate doit être :



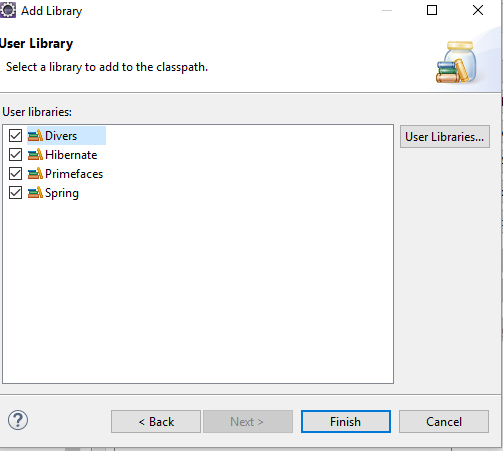
\* Ajouter les 04 User Librairies au niveau de votre CLASSPATH comme le montre l'écran suivant :



- Cliquer sur "Add Library..":

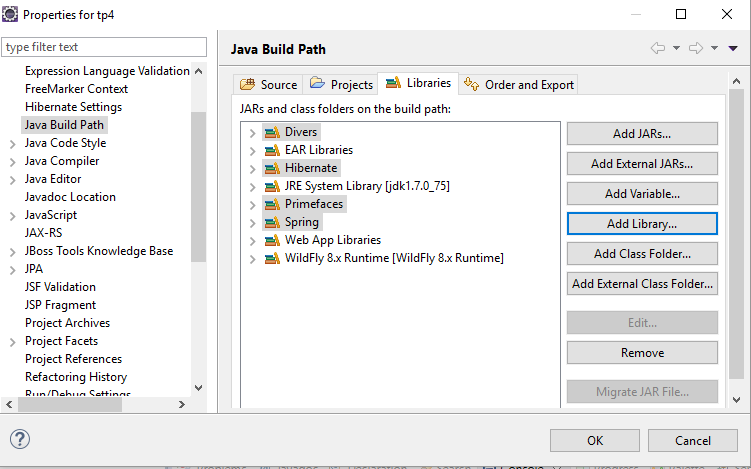


- Sélectionner "UserLibrary" et cliquer sur le bouton Next> :



- Cocher les 04 "User Library" e cliquer sur Finish.

- Vérifier que ces 4 "User Library" ont été bien ajoutées au niveau du CLASSPATH :



1. **Fichier jboss-deployment-structure.xml**

|  |
| --- |
| Créer le fichier jboss-deployment-structure.xml dans WebContent/WEB-INF : |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <jboss-deployment-structure>  <deployment>  <dependencies>  <module name=*"com.mysql"*/>  <module name=*"org.hibernate"*/>  <module name=*"org.spring"* meta-inf=*"export"* export=*"true"*/>  <module name=*"org.apache.commons.io"*/>  <module name=*"org.primefaces"* meta-inf=*"export"* export=*"true"*/>  </dependencies>  </deployment>  </jboss-deployment-structure> |

1. **Fichier jboss-web.xml**

|  |
| --- |
| Créer le fichier **jboss-web.xml** dans WebContent/WEB-INF : |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <jboss-web version=*"7.0"*  xmlns=*"http://www.jboss.com/xml/ns/javaee"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.jboss.com/xml/ns/javaee http://www.jboss.org/schema/jbossas/jboss-web\_7\_0.xsd"*>  <context-root>/tp4</context-root>  </jboss-web> |

1. **Fichier faces-config.xml**

|  |
| --- |
| Créer le fichier **faces-config.xml** dans WebContent/WEB-INF : |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <faces-config xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig\_2\_2.xsd"*  version=*"2.2"*>  <application>  <el-resolver>org.springframework.web.jsf.el.SpringBeanFacesELResolver</el-resolver>  <locale-config>  <default-locale>en</default-locale>  <supported-locale>fr</supported-locale>  <supported-locale>ar</supported-locale>  </locale-config>  <resource-bundle>  <base-name>messages</base-name>  <var>msg</var>  </resource-bundle>  </application>  </faces-config> |

1. **Fichier applicationContext.xml**

|  |
| --- |
| Créer le fichier **applicationContext.xml** dans WebContent/WEB-INF : |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans-3.2.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context-3.2.xsd*  *http://www.springframework.org/schema/aop*  *http://www.springframework.org/schema/aop/spring-aop-3.2.xsd*  *http://www.springframework.org/schema/tx*  *http://www.springframework.org/schema/tx/spring-tx-3.2.xsd"*>  <!--  <context:component-scan> permet de scanner un package et détecter automatiquement toutes les classes qui se trouvent au niveau de ce dernier  -->  <context:component-scan base-package=*"ma.gov.formation.cigma"* />    <!-- Pour accéder aux paramètres définis dans le fichier database.properties -->  <!-- le fichier database.properties doit être au niveau du classepath -->  <context:property-placeholder location=*"classpath:database.properties"* />    <!--  Le bean LocalContainerEntityManagerFactoryBean constitue la factory fournie par Spring qui permet de créer la session factory de JPA.  Ici le module qui est utilisé par Spring est Spring ORM.  Cette factory a besoin de :  1- datasource :le pool de connexion.  2- packagesToScan : pour permettre à Spring de détecter automatiquement toutes les classes du modèle.  3- jpaVendorAdapter :L'implémentation de JPA. Ici la classe qui est utilisée est HibernateJpaVendorAdapter qui encapsule Hibernate.    - La syntaxe : "<property name="dataSource" ref="dataSource" />" permet d'injecter le bean datasource dans l'attribut datasource moyennant le setter  (Injection par modifictaeur).    - Le champs "jpaProperties" (de type java.util.Properties) permet de configurer les propriétés de JPA. Dans cet exemple, les tables seront générées  automatiquement.    - Vous remarquez la syntaxe : ${hibernate.hbm2ddl.auto}, il s'agit de Spring EL (Spring Expression Language), il permet de récupérer la valeur  de la clé "hibernate.hbm2ddl.auto" qui se trouve au niveau du fichier database.properties.    - Dans la version 3.1 de Spring, le fichier persistence.xml a été abandonné.  -->    <bean id=*"sessionFactory"*  class=*"org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean"*>  <property name=*"dataSource"* ref=*"dataSource"* />  <property name=*"packagesToScan"* value=*"ma.gov.formation.cigma.service.modele"* />  <property name=*"jpaVendorAdapter"*>  <bean class=*"org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"* />  </property>  <property name=*"jpaProperties"*>  <props>  <prop key=*"hibernate.hbm2ddl.auto"*>${hibernate.hbm2ddl.auto}</prop>  <prop key=*"hibernate.dialect"*>${hibernate.dialect}</prop>  </props>  </property>  </bean>  <!--  - DriverManagerDataSource est un pool de connexion fourni par Spring (Le module Spring DAO). Pour configurer la datasource, il faut préciser le "driverClassName",  le "url", le "username" et le "password".  -->  <bean id=*"dataSource"*  class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>  <property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"* />  <property name=*"url"* value=*"${database.url}"* />  <property name=*"username"* value=*"${database.user}"* />  <property name=*"password"* value=*"${database.password}"* />  </bean>  <!--  - JpaTransactionManager est un intercepteur (proxy) qui permer de gérer les transactions. Ici le Design Pattern qui est implémenté est l'AOP (Aspect Oriented Programming).  Ce Proxy a besoin d'une instance de la Session Factory. Le bean "sessionFactory" défini ci-dessus sera injecté au niveau de l'attribut "entityManagerFactory".  -->  <bean id=*"transactionManager"* class=*"org.springframework.orm.jpa.JpaTransactionManager"*>  <property name=*"entityManagerFactory"* ref=*"sessionFactory"* />  </bean>    <!--  <tx:annotation-driven> permet au proxy "JpaTransactionManager" d'intercepter toutes les classes qui sont annotées par @Transactionnal.  -->  <tx:annotation-driven transaction-manager=*"transactionManager"* />  </beans> |

1. **Fichier web.xml**

|  |
| --- |
| **web.xml :** |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* version=*"3.1"*>  <display-name>tp2</display-name>  <welcome-file-list>  <welcome-file>/authentication/login.xhtml</welcome-file>  </welcome-file-list>    <!-- Add Support for Spring -->  <listener>  <listener-class>  org.springframework.web.context.ContextLoaderListener  </listener-class>  </listener>    <listener>  <listener-class>  org.springframework.web.context.request.RequestContextListener  </listener-class>  </listener>  <servlet>  <servlet-name>Faces Servlet</servlet-name>  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <context-param>  <param-name>primefaces.THEME</param-name>  <param-value>#{themeBean.selectedTheme}</param-value>  </context-param>  <servlet-mapping>  <servlet-name>Faces Servlet</servlet-name>  <url-pattern>\*.xhtml</url-pattern>  </servlet-mapping>  </web-app> |

1. **Fichier**

|  |
| --- |
| **src/database.properties** |
| database.driver=com.mysql.jdbc.Driver  database.url=jdbc:mysql://localhost:3306/formation\_db  database.user=root  database.password=root  hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect  hibernate.show\_sql=true  hibernate.hbm2ddl.auto=update |

1. **Le modèle :**

|  |
| --- |
| **La classe BaseEntity :** |
| **package** ma.gov.formation.cigma.service.modele;  **import** java.io.Serializable;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.Id;  **import** javax.persistence.MappedSuperclass;  @MappedSuperclass  **public** **abstract** **class** BaseEntity **implements** Serializable{  **private** **static** **final** **long** ***serialVersionUID*** = -663957177982798319L;  @GeneratedValue  @Id  **protected** Integer id;  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  } |

|  |
| --- |
| **La classe Article :** |
| **package** ma.gov.formation.cigma.service.modele;  **import** java.io.Serializable;  **import** javax.persistence.Basic;  **import** javax.persistence.CascadeType;  **import** javax.persistence.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.FetchType;  **import** javax.persistence.JoinColumn;  **import** javax.persistence.ManyToOne;  **import** javax.persistence.Table;  @Entity  @Table(name = "Article")  **public** **class** Article **extends** BaseEntity **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 6161046340508532663L;  @Column(name = "libelle")  @Basic(fetch = FetchType.***LAZY***)  **private** String designation;  **private** Double prix;  **private** Integer quantiteEnStock;  @ManyToOne(cascade = CascadeType.***ALL***)  @JoinColumn(name = "CAT\_ID", nullable = **false**)  **private** Categorie categorie = **new** Categorie();  **private** String photo;  @Override  **public** String toString() {  **return** "Article [id=" + id + ", designation=" + designation + ", prix="  + prix + ", quantiteEnStock=" + quantiteEnStock + "]";  }  **public** String getPhoto() {  **return** photo;  }  **public** **void** setPhoto(String photo) {  **this**.photo = photo;  }  **public** Categorie getCategorie() {  **return** categorie;  }  **public** **void** setCategorie(Categorie categorie) {  **this**.categorie = categorie;  }  **public** Article() {  }  **public** Article(String designation, Double prix, Integer quantiteEnStock) {  **super**();  **this**.designation = designation;  **this**.prix = prix;  **this**.quantiteEnStock = quantiteEnStock;  }    **public** Article(String designation, Double prix, Integer quantiteEnStock,String photo) {  **this**(designation, prix, quantiteEnStock);  **this**.photo = photo;  }  **public** Article(String designation, Double prix, Integer quantiteEnStock,  Categorie categorie, String photo) {  **super**();  **this**.designation = designation;  **this**.prix = prix;  **this**.quantiteEnStock = quantiteEnStock;  **this**.categorie = categorie;  **this**.photo = photo;  }  **public** Article(String designation, Double prix, Integer quantiteEnStock,  Categorie categorie) {  **this**(designation, prix, quantiteEnStock);  **this**.categorie = categorie;  }  **public** Article(Integer id, String designation, Double prix,  Integer quantiteEnStock, Categorie categorie) {  **this**(designation, prix, quantiteEnStock, categorie);  **this**.id = id;  }  **public** Article(Integer id) {  **this**.id = id;  }    **public** String getDesignation() {  **return** designation;  }  **public** **void** setDesignation(String designation) {  **this**.designation = designation;  }  **public** Double getPrix() {  **return** prix;  }  **public** **void** setPrix(Double prix) {  **this**.prix = prix;  }  **public** Integer getQuantiteEnStock() {  **return** quantiteEnStock;  }  **public** **void** setQuantiteEnStock(Integer quantiteEnStock) {  **this**.quantiteEnStock = quantiteEnStock;  }  } |

|  |
| --- |
| **La classe Categorie :** |
| **package** ma.gov.formation.cigma.service.modele;  **import** java.io.Serializable;  **import** java.util.ArrayList;  **import** java.util.List;  **import** javax.persistence.CascadeType;  **import** javax.persistence.Entity;  **import** javax.persistence.FetchType;  **import** javax.persistence.OneToMany;  @Entity  **public** **class** Categorie **extends** BaseEntity **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = -5342496290945790545L;  **private** String code;  **private** String libelle;    @OneToMany(mappedBy="categorie",fetch=FetchType.***LAZY***,cascade=CascadeType.***ALL***)  **private** List<Article> articles=**new** ArrayList<>();  **public** Categorie() {  **super**();  }  **public** Categorie(**int** id) {  **this**.id = id;  }    **public** Categorie(String code, String libelle) {  **super**();  **this**.code = code;  **this**.libelle = libelle;  }  **public** String getCode() {  **return** code;  }  **public** **void** setCode(String code) {  **this**.code = code;  }  **public** String getLibelle() {  **return** libelle;  }  **public** **void** setLibelle(String libelle) {  **this**.libelle = libelle;  }  **public** **void** addArticle(Article article) {  articles.add(article);  article.setCategorie(**this**);  }  } |

|  |
| --- |
| **La classe Theme :** |
| **package** ma.gov.formation.cigma.service.modele;  **import** java.io.Serializable;  **import** javax.persistence.Entity;  @Entity  **public** **class** Theme **extends** BaseEntity **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = -8553346591231350594L;  **private** String code;  **private** String libelle;  **private** String themeByDefault;    **public** String getThemeByDefault() {  **return** themeByDefault;  }  **public** **void** setThemeByDefault(String themeByDefault) {  **this**.themeByDefault = themeByDefault;  }  **public** Theme() {  }    **public** Theme(Integer id) {  **this**.id=id;  }    **public** Theme(String code, String libelle) {  **super**();  **this**.code = code;  **this**.libelle = libelle;  }    **public** Theme(String code, String libelle,String themeByDefault) {  **super**();  **this**.code = code;  **this**.libelle = libelle;  **this**.themeByDefault=themeByDefault;  }    **public** Theme(Integer id,String code, String libelle) {  **this**.id=id;  **this**.code = code;  **this**.libelle = libelle;  }  **public** String getCode() {  **return** code;  }  **public** **void** setCode(String code) {  **this**.code = code;  }  **public** String getLibelle() {  **return** libelle;  }  **public** **void** setLibelle(String libelle) {  **this**.libelle = libelle;  }  } |

|  |
| --- |
| **La classe User :** |
| **package** ma.gov.formation.cigma.service.modele;  **import** java.io.Serializable;  **import** java.util.ArrayList;  **import** java.util.List;  **import** javax.persistence.CascadeType;  **import** javax.persistence.Entity;  **import** javax.persistence.OneToMany;  @Entity  **public** **class** User **extends** BaseEntity **implements** Serializable{  **private** **static** **final** **long** ***serialVersionUID*** = 1091781481106770718L;  **private** String userName;  **private** String password;  @OneToMany(mappedBy="user",cascade = CascadeType.***ALL***)  **private** List<UserRole> roles=**new** ArrayList<>();    **public** User() {  **super**();  // **TODO** Auto-generated constructor stub  }    **public** User(Integer id) {  **this**.id = id;  }  **public** User(Integer id, String userName, String password, List<UserRole> roles) {  **super**();  **this**.id = id;  **this**.userName = userName;  **this**.password = password;  **this**.roles = roles;  }    **public** User(Integer id, String userName, String password) {  **super**();  **this**.id = id;  **this**.userName = userName;  **this**.password = password;  }    **public** User(String userName, String password, List<UserRole> roles) {  **super**();  **this**.userName = userName;  **this**.password = password;  **this**.roles = roles;  }  **public** User(String userName, String password) {  **super**();  **this**.userName = userName;  **this**.password = password;  }    **public** String getUserName() {  **return** userName;  }  **public** **void** setUserName(String userName) {  **this**.userName = userName;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  **public** List<UserRole> getRoles() {  **return** roles;  }  **public** **void** setRoles(List<UserRole> roles) {  **this**.roles = roles;  }  **public** **void** addUserRole(UserRole userRole) {  roles.add(userRole);  userRole.setUser(**this**);  }  } |

|  |
| --- |
| **La classe UserRole :** |
| **package** ma.gov.formation.cigma.service.modele;  **import** java.io.Serializable;  **import** javax.persistence.Entity;  **import** javax.persistence.JoinColumn;  **import** javax.persistence.ManyToOne;  @Entity  **public** **class** UserRole **extends** BaseEntity **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = -4769371604279552851L;  **private** String role;  **private** String libelle;  @ManyToOne  @JoinColumn(name="USER\_ID",nullable=**false**)  **private** User user;    **public** UserRole(Integer id) {  **this**.id = id;  }    **public** UserRole(String role, String libelle, User user) {  **super**();  **this**.role = role;  **this**.libelle = libelle;  **this**.user = user;  }  **public** UserRole(Integer id, String role, String libelle, User user) {  **super**();  **this**.id = id;  **this**.role = role;  **this**.libelle = libelle;  **this**.user = user;  }  **public** User getUser() {  **return** user;  }  **public** **void** setUser(User user) {  **this**.user = user;  }  **public** UserRole() {  **super**();  }    **public** UserRole(Integer id, String role, String libelle) {  **super**();  **this**.id = id;  **this**.role = role;  **this**.libelle = libelle;  }  **public** UserRole(String role, String libelle) {  **super**();  **this**.role = role;  **this**.libelle = libelle;  }  **public** String getRole() {  **return** role;  }  **public** **void** setRole(String role) {  **this**.role = role;  }  **public** String getLibelle() {  **return** libelle;  }  **public** **void** setLibelle(String libelle) {  **this**.libelle = libelle;  }    } |

1. **Les "Value Object"**

|  |
| --- |
| **La classe BaseVo :** |
| **package** ma.gov.formation.cigma.domaine;  **import** java.io.Serializable;  **public** **class** BaseVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 8500324480292121343L;  **protected** Integer id;  **protected** String code;  **protected** String libelle;  **public** BaseVo(Integer id) {  **super**();  **this**.id = id;  }  **public** BaseVo() {  **super**();  }  **public** BaseVo(String code, String libelle) {  **super**();  **this**.code = code;  **this**.libelle = libelle;  }  **public** BaseVo(Integer id, String code, String libelle) {  **super**();  **this**.id = id;  **this**.code = code;  **this**.libelle = libelle;  }  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getCode() {  **return** code;  }  **public** **void** setCode(String code) {  **this**.code = code;  }  **public** String getLibelle() {  **return** libelle;  }  **public** **void** setLibelle(String libelle) {  **this**.libelle = libelle;  }  } |

|  |
| --- |
| **La classe ArticleVo :** |
| **package** ma.gov.formation.cigma.domaine.article;  **import** java.io.Serializable;  **import** ma.gov.formation.cigma.domaine.BaseVo;  **import** ma.gov.formation.cigma.domaine.categorie.CategorieVo;  **public** **class** ArticleVo **extends** BaseVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 6161046340508532663L;  **private** String designation;  **private** Double prix;  **private** Integer quantiteEnStock;  **private** CategorieVo categorie=**new** CategorieVo();  **private** String photo;  **public** ArticleVo(Integer id) {  **super**();  **this**.id = id;  }  **public** String getPhoto() {  **return** photo;  }  **public** **void** setPhoto(String photo) {  **this**.photo = photo;  }  **public** CategorieVo getCategorie() {  **return** categorie;  }  **public** **void** setCategorie(CategorieVo categorie) {  **this**.categorie = categorie;  }  **private** **boolean** editable;  **public** **boolean** isEditable() {  **return** editable;  }  **public** **void** setEditable(**boolean** editable) {  **this**.editable = editable;  }  **public** ArticleVo() {  }  **public** ArticleVo(Integer id, String designation, Double prix,  Integer quantiteEnStock) {  **this**.id = id;  **this**.designation = designation;  **this**.prix = prix;  **this**.quantiteEnStock = quantiteEnStock;  categorie=**new** CategorieVo();  photo=**null**;  }    **public** ArticleVo(String designation, Double prix,  Integer quantiteEnStock,String photo) {  **this**.designation = designation;  **this**.prix = prix;  **this**.quantiteEnStock = quantiteEnStock;  **this**.photo=photo;  }    **public** **void** clear() {  id = **null**;  designation = **null**;  prix = **null**;  quantiteEnStock = **null**;  }  **public** String getDesignation() {  **return** designation;  }  **public** **void** setDesignation(String designation) {  **this**.designation = designation;  }  **public** Double getPrix() {  **return** prix;  }  **public** **void** setPrix(Double prix) {  **this**.prix = prix;  }  **public** Integer getQuantiteEnStock() {  **return** quantiteEnStock;  }  **public** **void** setQuantiteEnStock(Integer quantiteEnStock) {  **this**.quantiteEnStock = quantiteEnStock;  }  } |

|  |
| --- |
| **La classe ArticleConverter :** |
| **package** ma.gov.formation.cigma.domaine.article;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.categorie.CategoreConverter;  **import** ma.gov.formation.cigma.service.modele.Article;  **public** **class** ArticleConverter {  **public** **static** Article createModele(ArticleVo vo) {  Article modele = **new** Article();  modele.setId(vo.getId());  modele.setDesignation(vo.getDesignation());  modele.setPrix(vo.getPrix());  modele.setQuantiteEnStock(vo.getQuantiteEnStock());  modele.getCategorie().setCode(vo.getCategorie().getCode());  modele.setPhoto(vo.getPhoto());  **return** modele;  }  **public** **static** ArticleVo createArticleVo(Article modele) {  ArticleVo vo = **new** ArticleVo();  vo.setId(modele.getId());  vo.setDesignation(modele.getDesignation());  vo.setPrix(modele.getPrix());  vo.setQuantiteEnStock(modele.getQuantiteEnStock());  vo.setCategorie(CategoreConverter.*createCategorieVo*(modele.getCategorie()));  vo.setPhoto(modele.getPhoto());  **return** vo;  }  **public** **static** List<ArticleVo> getArticleVoList(List<Article> modeles) {  List<ArticleVo> vos = **new** ArrayList<>();  **if** (modeles == **null** || modeles.isEmpty())  **return** **null**;  **for** (Article modele : modeles) {  vos.add(*createArticleVo*(modele));  }  **return** vos;  }  } |

|  |
| --- |
| **La classe CategorieVo :** |
| **package** ma.gov.formation.cigma.domaine.categorie;  **import** java.io.Serializable;  **import** ma.gov.formation.cigma.domaine.BaseVo;  **public** **class** CategorieVo **extends** BaseVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 6901655927457141058L;  **public** CategorieVo() {  **super**();  }  **public** CategorieVo(Integer id) {  **super**();  **this**.id = id;  }  **public** CategorieVo(String code,String libelle) {  **super**(code,libelle);  }  } |

|  |
| --- |
| **La classe CategoreConverter :** |
| **package** ma.gov.formation.cigma.domaine.categorie;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.Categorie;  **public** **class** CategoreConverter {  **public** **static** Categorie createModele(CategorieVo vo) {  Categorie modele = **new** Categorie();  modele.setId(vo.getId());  modele.setCode(vo.getCode());  modele.setLibelle(vo.getLibelle());  **return** modele;  }  **public** **static** CategorieVo createCategorieVo(Categorie modele) {  CategorieVo vo = **new** CategorieVo();  vo.setId(modele.getId());  vo.setCode(modele.getCode());  vo.setLibelle(modele.getLibelle());  **return** vo;  }    **public** **static** List<CategorieVo> getCategorieVoList(List<Categorie> modeles) {  List<CategorieVo> vos=**new** ArrayList<>();  **if** (modeles == **null** || modeles.isEmpty())  **return** **null**;  **for** (Categorie modele : modeles) {  vos.add(*createCategorieVo*(modele));  }  **return** vos;  }  } |

|  |
| --- |
| **La classe ThemeVo :** |
| **package** ma.gov.formation.cigma.domaine.theme;  **import** ma.gov.formation.cigma.domaine.BaseVo;  **public** **class** ThemeVo **extends** BaseVo{  **private** **static** **final** **long** ***serialVersionUID*** = -8884029347619819836L;  **private** String themeByDefault;  **public** String getThemeByDefault() {  **return** themeByDefault;  }  **public** **void** setThemeByDefault(String themeByDefault) {  **this**.themeByDefault = themeByDefault;  }  **public** ThemeVo() {  }  **public** ThemeVo(**int** id, String code, String libelle) {  **super**(id,code,libelle);  }    **public** ThemeVo(**int** id, String code, String libelle,String themeByDefault) {  **super**(id,code,libelle);  **this**.themeByDefault=themeByDefault;  }    **public** ThemeVo(String code, String libelle,String themeByDefault) {  **super**(code,libelle);  **this**.themeByDefault=themeByDefault;  }    **public** ThemeVo(String code, String libelle) {  **super**(code,libelle);  **this**.themeByDefault=**null**;  }  } |

|  |
| --- |
| **La classe ThemeConverter :** |
| **package** ma.gov.formation.cigma.domaine.theme;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.Theme;  **public** **class** ThemeConverter {  **public** **static** Theme createModele(ThemeVo vo) {  Theme modele = **new** Theme();  modele.setCode(vo.getCode());  modele.setLibelle(vo.getLibelle());  modele.setId(vo.getId());  modele.setThemeByDefault(vo.getThemeByDefault());  **return** modele;  }  **public** **static** ThemeVo createThemeVo(Theme modele) {  ThemeVo vo = **new** ThemeVo();  vo.setId(modele.getId());  vo.setCode(modele.getCode());  vo.setLibelle(modele.getLibelle());  vo.setThemeByDefault(modele.getThemeByDefault());  **return** vo;  }  **public** **static** List<ThemeVo> getThemeVoList(List<Theme> modeles) {  List<ThemeVo> vos = **new** ArrayList<>();  **if** (modeles == **null** || modeles.isEmpty())  **return** **null**;  **for** (Theme modele : modeles) {  vos.add(*createThemeVo*(modele));  }  **return** vos;  }  } |

|  |
| --- |
| **La classe UserVo :** |
| **package** ma.gov.formation.cigma.domaine.user;  **import** java.io.Serializable;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.BaseVo;  **import** ma.gov.formation.cigma.presentation.Constants;  **public** **class** UserVo **extends** BaseVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = -5500475381869205248L;  **private** String userName;  **private** String password;  **private** List<UserRoleVo> roles = **new** ArrayList<>();    **public** UserVo(Integer id) {  **super**();  **this**.id = id;  }  **public** **void** setAdmin(**boolean** admin) {  **this**.admin = admin;  }  **public** **void** setClient(**boolean** client) {  **this**.client = client;  }  **private** **boolean** admin;  **private** **boolean** client;  **public** **boolean** isAdmin() {  **return** haseRole(Constants.***ADMIN\_ROLE***);  }  **public** **boolean** isClient() {  **return** haseRole(Constants.***CLIENT\_ROLE***);  }  **public** List<UserRoleVo> getRoles() {  **return** roles;  }  **public** **void** setRoles(List<UserRoleVo> roles) {  **this**.roles = roles;  }  **public** UserVo() {  **super**();  }  **public** UserVo(String userName, String password) {  **super**();  **this**.userName = userName;  **this**.password = password;  }  **public** UserVo(Integer id, String userName, String password) {  **super**();  **this**.id = id;  **this**.userName = userName;  **this**.password = password;  }  **public** String getUserName() {  **return** userName;  }  **public** **void** setUserName(String userName) {  **this**.userName = userName;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  **public** **boolean** haseRole(String role) {  **boolean** result = **false**;  **if** (roles.isEmpty())  **return** **false**;  **for** (UserRoleVo vo : roles) {  **if** (role.equals(vo.getCode()))  **return** **true**;  }  **return** result;  }  } |

|  |
| --- |
| **La classe UserRoleVo :** |
| **package** ma.gov.formation.cigma.domaine.user;  **import** java.io.Serializable;  **import** ma.gov.formation.cigma.domaine.BaseVo;  **public** **class** UserRoleVo **extends** BaseVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 4928414063636905875L;  **public** UserRoleVo() {  **super**();  }  **public** UserRoleVo(String code, String libelle) {  **super**();  **this**.code = code;  **this**.libelle = libelle;  }  **public** UserRoleVo(Integer id, String code, String libelle) {  **this**(code,libelle);  **this**.id = id;  }  } |

|  |
| --- |
| **La classe UserConverter :** |
| **package** ma.gov.formation.cigma.domaine.user.converter;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.user.UserRoleVo;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **import** ma.gov.formation.cigma.service.modele.User;  **public** **class** UserConverter {  **public** **static** User createModele(UserVo vo) {  User modele = **new** User();  modele.setId(vo.getId());  modele.setUserName(vo.getUserName());  modele.setPassword(vo.getPassword());  List<UserRoleVo> roles=vo.getRoles();  **for** (UserRoleVo userRoleVo : roles) {  modele.addUserRole(UserRoleConverter.*createModele*(userRoleVo));  }  **return** modele;  }  **public** **static** UserVo createUserVo(User modele) {  UserVo vo = **null**;  **if** (modele == **null**)  **return** vo;  vo = **new** UserVo();  vo.setId(modele.getId());  vo.setUserName(modele.getUserName());  vo.setPassword(modele.getPassword());  **if** (modele.getRoles() == **null** || modele.getRoles().isEmpty())  **return** vo;  vo.setRoles(UserRoleConverter.*getUserRoleVoList*(modele.getRoles()));  **return** vo;  }  **public** **static** List<UserVo> getUserVoList(List<User> modeles) {  List<UserVo> vos = **new** ArrayList<>();  **if** (modeles == **null** || modeles.isEmpty())  **return** **null**;  **for** (User modele : modeles) {  vos.add(*createUserVo*(modele));  }  **return** vos;  }  } |

|  |
| --- |
| **La classe UserRoleConverter :** |
| **package** ma.gov.formation.cigma.domaine.user.converter;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.user.UserRoleVo;  **import** ma.gov.formation.cigma.service.modele.UserRole;  **public** **class** UserRoleConverter {  **public** **static** UserRole createModele(UserRoleVo vo) {  UserRole modele = **new** UserRole();  modele.setId(vo.getId());  modele.setLibelle(vo.getLibelle());  modele.setRole(vo.getCode());  **return** modele;  }  **public** **static** UserRoleVo createUserRoleVo(UserRole modele) {  UserRoleVo vo = **new** UserRoleVo();  vo.setId(modele.getId());  vo.setLibelle(modele.getLibelle());  vo.setCode(modele.getRole());  **return** vo;  }  **public** **static** List<UserRoleVo> getUserRoleVoList(List<UserRole> modeles) {  List<UserRoleVo> vos = **new** ArrayList<>();  **if** (modeles == **null** || modeles.isEmpty())  **return** vos;  **for** (UserRole modele : modeles) {  vos.add(*createUserRoleVo*(modele));  }  **return** vos;  }  } |

1. **La couche DAO**

|  |
| --- |
| **L'interface IBaseDao :** |
| **package** ma.gov.formation.cigma.dao;  **import** java.util.List;  **public** **interface** IBaseDao {  <T> T save(T t) **throws** Exception;  <T> T getById (T t) **throws** Exception;  <T> List<T> getAll(T t) **throws** Exception;  <T> **void** remove(T t) **throws** Exception;  } |

|  |
| --- |
| **L'interface IArticleDao :** |
| **package** ma.gov.formation.cigma.dao;  **import** ma.gov.formation.cigma.service.modele.Categorie;  **public** **interface** IArticleDao **extends** IBaseDao{  Categorie getCategoreByCode(String code) **throws** Exception;  } |

|  |
| --- |
| **L'interface IFileDao :** |
| **package** ma.gov.formation.cigma.dao;  **import** java.io.File;  **import** java.io.IOException;  **import** java.io.InputStream;  **public** **interface** IFileDao {  **void** copy(InputStream input, File fileDest) **throws** IOException;  } |

|  |
| --- |
| **L'interface IThemeDao :** |
| **package** ma.gov.formation.cigma.dao;  **import** ma.gov.formation.cigma.service.modele.Theme;  **public** **interface** IThemeDao **extends** IBaseDao {  Theme getDefaultTheme() **throws** Exception;  Theme getByCode(String code) **throws** Exception;  } |

|  |
| --- |
| **L'interface IUserDao :** |
| **package** ma.gov.formation.cigma.dao;  **import** ma.gov.formation.cigma.service.modele.User;  **public** **interface** IUserDao **extends** IBaseDao {  User getByLogin(String login) **throws** Exception;  } |

|  |
| --- |
| **La classe BaseDaoImpl :** |
| **package** ma.gov.formation.cigma.dao.impl;  **import** java.util.List;  **import** javax.inject.Named;  **import** javax.persistence.EntityManager;  **import** javax.persistence.PersistenceContext;  **import** ma.gov.formation.cigma.dao.IBaseDao;  **import** ma.gov.formation.cigma.service.modele.BaseEntity;  @Named("baseDao")  **public** **class** BaseDaoImpl **implements** IBaseDao {    @PersistenceContext  **private** EntityManager session;  **public** EntityManager getSession() {  **return** session;  }  **public** **void** setSession(EntityManager session) {  **this**.session = session;  }  @Override  **public** <T> T save(T t) **throws** Exception {  **return** session.merge(t);  }  @SuppressWarnings("unchecked")  @Override  **public** <T> T getById(T t) **throws** Exception {  String entityName = t.getClass().getName();  **int** id = ((BaseEntity) t).getId();  String req = "from " + entityName + " where id=" + id;  **return** (T) session.createQuery(req).getSingleResult();  }  @SuppressWarnings("unchecked")  @Override  **public** <T> List<T> getAll(T t) **throws** Exception {  String entityName = t.getClass().getName();  String req = "from " + entityName;  **return** session.createQuery(req).getResultList();  }  @Override  **public** <T> **void** remove(T t) **throws** Exception {  T tp = getById(t);  **if** (tp != **null**)  session.remove(tp);  }  } |

|  |
| --- |
| **La classe ArticleDaoImpl :** |
| **package** ma.gov.formation.cigma.dao.impl;  **import** java.util.List;  **import** javax.inject.Named;  **import** javax.persistence.Query;  **import** ma.gov.formation.cigma.dao.IArticleDao;  **import** ma.gov.formation.cigma.service.modele.Categorie;  @Named("articleDao")  **public** **class** ArticleDaoImpl **extends** BaseDaoImpl **implements** IArticleDao {  @SuppressWarnings("unchecked")  @Override  **public** Categorie getCategoreByCode(String code) **throws** Exception {  List<Categorie> liste = **null**;  Query q = getSession()  .createQuery("select c from Categorie c where c.code=:code");  q.setParameter("code", code);  liste = (List<Categorie>) q.getResultList();  **return** liste == **null** || liste.isEmpty() ? **null** : liste.get(0);  }  } |

|  |
| --- |
| **La classe FileDaoImpl** : |
| **package** ma.gov.formation.cigma.dao.impl;  **import** java.io.File;  **import** java.io.FileOutputStream;  **import** java.io.IOException;  **import** java.io.InputStream;  **import** java.io.OutputStream;  **import** ma.gov.formation.cigma.dao.IFileDao;  **import** org.apache.commons.io.IOUtils;  **public** **class** FileDaoImpl **implements** IFileDao {  @Override  **public** **void** copy(InputStream input, File fileDest) **throws** IOException {  OutputStream outputStream= **new** FileOutputStream(fileDest);  IOUtils.*copy*(input, outputStream);  }  } |

|  |
| --- |
| **La classe ThemeDaoImpl :** |
| **package** ma.gov.formation.cigma.dao.impl;  **import** java.util.List;  **import** javax.inject.Named;  **import** javax.persistence.Query;  **import** ma.gov.formation.cigma.dao.IThemeDao;  **import** ma.gov.formation.cigma.service.modele.Theme;  @Named("themeDao")  **public** **class** ThemeDaoImpl **extends** BaseDaoImpl **implements** IThemeDao {  @Override  @SuppressWarnings("unchecked")  **public** Theme getDefaultTheme() **throws** Exception {  Theme a = **null**;  Query q = getSession().createQuery("from Theme t where t.themeByDefault=:themeByDefault");  q.setParameter("themeByDefault", "YES");  List<Theme> result = q.getResultList();  a = (result != **null** && !result.isEmpty()) ? result.get(0) : **null**;  **return** a;  }  @Override  **public** Theme getByCode(String code) **throws** Exception {  Theme a = **null**;  Query q = getSession().createQuery("from Theme t where t.code=:code");  q.setParameter("code", code);  List<Theme> result = q.getResultList();  a = (result != **null** && !result.isEmpty()) ? result.get(0) : **null**;  **return** a;  }  } |

|  |
| --- |
| **La classe UserDaoImpl :** |
| **package** ma.gov.formation.cigma.dao.impl;  **import** java.util.List;  **import** javax.inject.Named;  **import** javax.persistence.Query;  **import** ma.gov.formation.cigma.dao.IUserDao;  **import** ma.gov.formation.cigma.service.modele.User;  @Named("userDao")  **public** **class** UserDaoImpl **extends** BaseDaoImpl **implements** IUserDao {  @Override  @SuppressWarnings("unchecked")  **public** User getByLogin(String userName) **throws** Exception {  User a = **null**;  Query q = getSession().createQuery("from User u where u.userName=:userName");  q.setParameter("userName", userName);  List<User> result = q.getResultList();  a = (result != **null** && !result.isEmpty()) ? result.get(0) : **null**;  **return** a;  }  } |

1. **La couche Service**

|  |
| --- |
| **L'interface IBaseService :** |
| **package** ma.gov.formation.cigma.service;  **import** java.util.List;  **public** **interface** IBaseService {  <T> T getById (T t) **throws** Exception;  <T> List<T> getAll(T t) **throws** Exception;  <T> **void** remove(T t) **throws** Exception;  <T> **void** save(T t) **throws** Exception;  } |

|  |
| --- |
| **L'interface IArticleService :** |
| **package** ma.gov.formation.cigma.service;  **import** ma.gov.formation.cigma.service.modele.Categorie;  **public** **interface** IArticleService **extends** IBaseService {  Categorie getCategoreByCode(String code) **throws** Exception;  } |

|  |
| --- |
| **L'interface IFileService :** |
| **package** ma.gov.formation.cigma.service;  **import** java.io.IOException;  **import** java.io.InputStream;  **public** **interface** IFileService {  **void** copyPhotoArticle(InputStream inputstream, String fileName) **throws** IOException;  } |

|  |
| --- |
| **L'interface IThemeService :** |
| **package** ma.gov.formation.cigma.service;  **import** ma.gov.formation.cigma.domaine.theme.ThemeVo;  **public** **interface** IThemeService **extends** IBaseService {  **public** ThemeVo getDefaultTheme() **throws** Exception;  **public** ThemeVo updateThemeByDefault(String code) **throws** Exception;  } |

|  |
| --- |
| **L'interface IUserService :** |
| **package** ma.gov.formation.cigma.service;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **public** **interface** IUserService **extends** IBaseService{  **public** UserVo getByLogin(String login) **throws** Exception;  } |

|  |
| --- |
| **La classe BaseServiceImpl :** |
| **package** ma.gov.formation.cigma.service.impl;  **import** java.util.List;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.dao.IBaseDao;  **import** ma.gov.formation.cigma.domaine.article.ArticleConverter;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  **import** ma.gov.formation.cigma.domaine.categorie.CategoreConverter;  **import** ma.gov.formation.cigma.domaine.categorie.CategorieVo;  **import** ma.gov.formation.cigma.domaine.theme.ThemeConverter;  **import** ma.gov.formation.cigma.domaine.theme.ThemeVo;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **import** ma.gov.formation.cigma.domaine.user.converter.UserConverter;  **import** ma.gov.formation.cigma.service.IBaseService;  **import** ma.gov.formation.cigma.service.modele.Article;  **import** ma.gov.formation.cigma.service.modele.Categorie;  **import** ma.gov.formation.cigma.service.modele.Theme;  **import** ma.gov.formation.cigma.service.modele.User;  **import** org.springframework.transaction.annotation.Transactional;  @Named("baseService")  @Transactional  **public** **class** BaseServiceImpl **implements** IBaseService {  @Inject  **private** IBaseDao baseDao;  **public** IBaseDao getBaseDao() {  **return** baseDao;  }  **public** **void** setBaseDao(IBaseDao baseDao) {  **this**.baseDao = baseDao;  }  @SuppressWarnings("unchecked")  @Override  **public** <T> T getById(T t) **throws** Exception {  T result = **null**;  **if** (t **instanceof** ArticleVo) {  Integer id = ((ArticleVo) t).getId();  result = (T) ArticleConverter.*createArticleVo*(baseDao  .getById(**new** Article(id)));  }  **if** (t **instanceof** UserVo) {  Integer id = ((UserVo) t).getId();  result = (T) UserConverter.*createUserVo*(baseDao  .getById(**new** User(id)));  }  **if** (t **instanceof** CategorieVo) {  Integer id = ((CategorieVo) t).getId();  result = (T) CategoreConverter.*createCategorieVo*(baseDao  .getById(**new** Categorie(id)));  }    **if** (t **instanceof** ThemeVo) {  Integer id = ((ThemeVo) t).getId();  result = (T) ThemeConverter.*createThemeVo*(baseDao  .getById(**new** Theme(id)));  }  **return** result;  }  @SuppressWarnings("unchecked")  @Override  **public** <T> List<T> getAll(T t) **throws** Exception {  List<T> list = **null**;  **if** (t **instanceof** ArticleVo)  list = (List<T>) ArticleConverter.*getArticleVoList*(baseDao  .getAll(**new** Article()));  **if** (t **instanceof** UserVo)  list = (List<T>) UserConverter.*getUserVoList*(baseDao  .getAll(**new** User()));  **if** (t **instanceof** CategorieVo)  list = (List<T>) CategoreConverter.*getCategorieVoList*(baseDao  .getAll(**new** Categorie()));    **if** (t **instanceof** ThemeVo)  list = (List<T>) ThemeConverter.*getThemeVoList*(baseDao  .getAll(**new** Theme()));  **return** list;  }  @Override  **public** <T> **void** remove(T t) **throws** Exception {  **if** (t **instanceof** ArticleVo)  baseDao.remove(ArticleConverter.*createModele*((ArticleVo) t));  **if** (t **instanceof** UserVo)  baseDao.remove(UserConverter.*createModele*((UserVo) t));  **if** (t **instanceof** CategorieVo)  baseDao.remove(CategoreConverter.*createModele*((CategorieVo) t));    **if** (t **instanceof** ThemeVo)  baseDao.remove(ThemeConverter.*createModele*((ThemeVo) t));  }  @Override  **public** <T> **void** save(T t) **throws** Exception {  **if** (t **instanceof** ArticleVo)  baseDao.save(ArticleConverter.*createModele*((ArticleVo) t));  **if** (t **instanceof** UserVo)  baseDao.save(UserConverter.*createModele*((UserVo) t));  **if** (t **instanceof** CategorieVo)  baseDao.save(CategoreConverter.*createModele*((CategorieVo) t));    **if** (t **instanceof** ThemeVo)  baseDao.save(ThemeConverter.*createModele*((ThemeVo) t));  }  } |

|  |
| --- |
| **La classe ArticleServiceImpl :** |
| **package** ma.gov.formation.cigma.service.impl;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.dao.IArticleDao;  **import** ma.gov.formation.cigma.domaine.article.ArticleConverter;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  **import** ma.gov.formation.cigma.service.IArticleService;  **import** ma.gov.formation.cigma.service.modele.Article;  **import** ma.gov.formation.cigma.service.modele.Categorie;  **import** org.springframework.transaction.annotation.Transactional;  @Named("articleService")  @Transactional  **public** **class** ArticleServiceImpl **extends** BaseServiceImpl **implements**  IArticleService {  @Inject  **private** IArticleDao articleDao;  **public** IArticleDao getArticleDao() {  **return** articleDao;  }  **public** **void** setArticleDao(IArticleDao articleDao) {  **this**.articleDao = articleDao;  }  @Override  **public** <T> **void** save(T t) **throws** Exception {  ArticleVo vo = (ArticleVo) t;  String code = vo.getCategorie().getCode();  Categorie catP = getCategoreByCode(code);  Article article = ArticleConverter.*createModele*(vo);  article.setCategorie(catP);  articleDao.save(article);  }  **public** Categorie getCategoreByCode(String code) **throws** Exception {  // **TODO** Auto-generated method stub  **return** articleDao.getCategoreByCode(code);  }  } |

|  |
| --- |
| **La classe FileServiceImpl :** |
| **package** ma.gov.formation.cigma.service.impl;  **import** java.io.File;  **import** java.io.IOException;  **import** java.io.InputStream;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.dao.IFileDao;  **import** ma.gov.formation.cigma.presentation.Constants;  **import** ma.gov.formation.cigma.service.IFileService;  @Named("fileService")  **public** **class** FileServiceImpl **implements** IFileService {  @Inject  **private** IFileDao fileDo;  **public** IFileDao getFileDo() {  **return** fileDo;  }  **public** **void** setFileDo(IFileDao fileDo) {  **this**.fileDo = fileDo;  }  @Override  **public** **void** copyPhotoArticle(InputStream inputstream, String fileName)  **throws** IOException {  File dir = **new** File(Constants.***IMAGES\_ARTICLES\_PATH***, fileName);  fileDo.copy(inputstream, dir);  }  } |

|  |
| --- |
| **La classe ThemeServiceImpl :** |
| **package** ma.gov.formation.cigma.service.impl;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.dao.IThemeDao;  **import** ma.gov.formation.cigma.domaine.theme.ThemeConverter;  **import** ma.gov.formation.cigma.domaine.theme.ThemeVo;  **import** ma.gov.formation.cigma.service.IThemeService;  **import** ma.gov.formation.cigma.service.modele.Theme;  **import** org.springframework.transaction.annotation.Transactional;  @Named("themeService")  @Transactional  **public** **class** ThemeServiceImpl **extends** BaseServiceImpl **implements** IThemeService {  @Inject  **private** IThemeDao themeDao;  **public** IThemeDao getThemeDao() {  **return** themeDao;  }  **public** **void** setThemeDao(IThemeDao themeDao) {  **this**.themeDao = themeDao;  }  @Override  **public** ThemeVo getDefaultTheme() **throws** Exception {  ThemeVo vo = **null**;  vo = ThemeConverter.*createThemeVo*(themeDao.getDefaultTheme());  **return** vo;  }  @Override  **public** ThemeVo updateThemeByDefault(String code) **throws** Exception {  ThemeVo vo=**null**;  Theme themeByDefaultInit=themeDao.getDefaultTheme();  themeByDefaultInit.setThemeByDefault(**null**);  Theme newThemeByDefault=themeDao.getByCode(code);  newThemeByDefault.setThemeByDefault("YES");  vo = ThemeConverter.*createThemeVo*(newThemeByDefault);  **return** vo;  }  } |

|  |
| --- |
| **La classe UserServiceImpl :** |
| **package** ma.gov.formation.cigma.service.impl;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.dao.IUserDao;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **import** ma.gov.formation.cigma.domaine.user.converter.UserConverter;  **import** ma.gov.formation.cigma.service.IUserService;  **import** org.springframework.transaction.annotation.Transactional;  @Named("userService")  @Transactional  **public** **class** UserServiceImpl **extends** BaseServiceImpl **implements** IUserService {  @Inject  **private** IUserDao userDao;  @Override  **public** UserVo getByLogin(String login) **throws** Exception {  UserVo vo = **null**;  vo = UserConverter.*createUserVo*(userDao.getByLogin(login));  **return** vo;  }  **public** IUserDao getUserDao() {  **return** userDao;  }  **public** **void** setUserDao(IUserDao userDao) {  **this**.userDao = userDao;  }  } |

1. **La couche présentation**

|  |
| --- |
| **La classe Utils :** |
| **package** ma.gov.formation.cigma.presentation;  **import** javax.faces.context.FacesContext;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpSession;  **public** **class** Utils {  **public** **static** HttpSession getSession() {  **return** (HttpSession) FacesContext.*getCurrentInstance*()  .getExternalContext().getSession(**false**);  }  **public** **static** HttpServletRequest getRequest() {  **return** (HttpServletRequest) FacesContext.*getCurrentInstance*()  .getExternalContext().getRequest();  }  } |

|  |
| --- |
| **La classe Constants :** |
| **package** ma.gov.formation.cigma.presentation;  **public** **interface** Constants {  **public** **static** **final** String ***ADMIN\_ROLE***="ADMIN";  **public** **static** **final** String ***CLIENT\_ROLE***="CLIENT";  **public** **static** **final** String ***ADMIN\_ROLE\_DESCRIPTION***="L'administrateur";  **public** **static** **final** String ***CLIENT\_ROLE\_DESCRIPTION***="Le client";  **public** **static** **final** String ***ERROR\_MSG***="ERROR\_MSG";  **public** **static** **final** String ***INFO\_MSG***="INFO\_MSG";  **public** **static** **final** String ***IMAGES\_ARTICLES\_PATH*** = "c:/photos\_articles/";  } |

|  |
| --- |
| **La classe BaseAction:** |
| **package** ma.gov.formation.cigma.presentation;  **import** java.util.ResourceBundle;  **import** javax.faces.application.Application;  **import** javax.faces.application.FacesMessage;  **import** javax.faces.context.FacesContext;  **public** **class** BaseAction {  **public** **void** addMessage(String keyMessage, String severity) {  FacesMessage message = **null**;  message = **new** FacesMessage(getMessage(keyMessage));  **if** (severity.equals(Constants.***ERROR\_MSG***))  message.setSeverity(FacesMessage.***SEVERITY\_ERROR***);  **if** (severity.equals(Constants.***INFO\_MSG***))  message.setSeverity(FacesMessage.***SEVERITY\_INFO***);  FacesContext.*getCurrentInstance*().addMessage(**null**, message);  }  **public** String toDo() {  **return** "/toDo.xhtml";  }  **public** String getMessage(String key) {  FacesContext context = FacesContext.*getCurrentInstance*();  Application app = context.getApplication();  ResourceBundle msg = app.getResourceBundle(context, "msg");  **return** msg.getString(key);  }  } |

|  |
| --- |
| **La classe ArticleBean :** |
| **package** ma.gov.formation.cigma.presentation.article;  **import** java.io.IOException;  **import** java.util.ArrayList;  **import** java.util.List;  **import** javax.annotation.PostConstruct;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  **import** ma.gov.formation.cigma.domaine.categorie.CategorieVo;  **import** ma.gov.formation.cigma.presentation.BaseAction;  **import** ma.gov.formation.cigma.presentation.Constants;  **import** ma.gov.formation.cigma.service.IArticleService;  **import** ma.gov.formation.cigma.service.IFileService;  **import** org.primefaces.event.RowEditEvent;  **import** org.primefaces.model.UploadedFile;  **import** org.springframework.context.annotation.Scope;  @Named("articleBean")  @Scope("request")  **public** **class** ArticleBean **extends** BaseAction {  **private** List<CategorieVo> categories;  @Inject  **private** IArticleService articleService;  **private** List<ArticleVo> articlesVo = **new** ArrayList<>();  **private** ArticleVo articleVo = **new** ArticleVo();  @Inject  **private** IFileService fileService;  **private** UploadedFile file;  **public** UploadedFile getFile() {  **return** file;  }  **public** **void** setFile(UploadedFile file) {  **this**.file = file;  }  @PostConstruct  **public** **void** init() {  **try** {  articlesVo = articleService.getAll(**new** ArticleVo());  categories = articleService.getAll(**new** CategorieVo());  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  }  }  **public** List<CategorieVo> getCategories() {  **return** categories;  }  **public** ArticleVo getArticleVo() {  **return** articleVo;  }  **public** **void** setArticleVo(ArticleVo articleVo) {  **this**.articleVo = articleVo;  }  **public** **void** setArticlesVo(List<ArticleVo> articlesVo) {  **this**.articlesVo = articlesVo;  }  **public** List<ArticleVo> getArticlesVo() {  **return** articlesVo;  }  **public** String addArticle() {  **try** {  fileService.copyPhotoArticle(file.getInputstream(),  file.getFileName());  articleVo.setPhoto(Constants.***IMAGES\_ARTICLES\_PATH***  + file.getFileName());  articleService.save(articleVo);  articlesVo = articleService.getAll(**new** ArticleVo());  articleVo.clear();  addMessage("article.ajoute.avec.succes", Constants.***INFO\_MSG***);  **return** **null**;  } **catch** (IOException e) {  addMessage("probleme.copie.fichier", Constants.***ERROR\_MSG***);  e.printStackTrace();  **return** **null**;  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  **return** **null**;  }  }  **public** String removeArticle(Integer id) {  **try** {  articleService.remove(id);  articlesVo = articleService.getAll(**new** ArticleVo());  articleVo.clear();  addMessage("article.supprime.avec.succes", Constants.***INFO\_MSG***);  **return** **null**;  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  **return** **null**;  }  }  **public** String list() {  **return** "/admin/article/articles.xhtml";  }  **public** **void** onRowEdit(RowEditEvent event) {  **try** {  articleVo = (ArticleVo) event.getObject();  addMessage("article.modifier.avec.succes", Constants.***INFO\_MSG***);  articleService.save(articleVo);  articlesVo = articleService.getAll(**new** ArticleVo());  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  }  }  **public** **void** onRowCancel(RowEditEvent event) {  addMessage("modifications.annulees", Constants.***INFO\_MSG***);  }  } |

|  |
| --- |
| **La classe LoginBean :** |
| **package** ma.gov.formation.cigma.presentation.authentication;  **import** javax.faces.application.ConfigurableNavigationHandler;  **import** javax.faces.context.FacesContext;  **import** javax.faces.event.ComponentSystemEvent;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** javax.servlet.http.HttpSession;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **import** ma.gov.formation.cigma.presentation.BaseAction;  **import** ma.gov.formation.cigma.presentation.Constants;  **import** ma.gov.formation.cigma.presentation.Utils;  **import** ma.gov.formation.cigma.service.IUserService;  **import** org.springframework.context.annotation.Scope;  @Named("loginBean")  @Scope("session")  **public** **class** LoginBean **extends** BaseAction {  @Inject  **private** IUserService userService;  **public** IUserService getUserService() {  **return** userService;  }  **public** **void** setUserService(IUserService userService) {  **this**.userService = userService;  }  **private** UserVo userVo = **new** UserVo();  **public** UserVo getUserVo() {  **return** userVo;  }  // logout event, invalidate session  **public** String logout() {  HttpSession session = Utils.*getSession*();  session.invalidate();  **return** "/authentication/login";  }  **public** **void** setUserVo(UserVo userVo) {  **this**.userVo = userVo;  }  **public** String login() {  UserVo userVoP;  **try** {  userVoP = userService.getByLogin(userVo.getUserName());  **if** (userVoP == **null**) {  addMessage("login.doesnt.exist", Constants.***ERROR\_MSG***);  **return** "/authentication/login";  }  **if** (!userVo.getPassword().equals(userVoP.getPassword())) {  addMessage("password.incorrect", Constants.***ERROR\_MSG***);  **return** "/authentication/login";  }  userVo = userVoP;  HttpSession session = Utils.*getSession*();  session.setAttribute("user", userVo);  **return** "/mainTemplate/mainTemplate";  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  **return** "/authentication/login";  }  }  **public** **void** isAdmin(ComponentSystemEvent event) {  FacesContext fc = FacesContext.*getCurrentInstance*();  **if** (Utils.*getSession*().getAttribute("user") == **null**) {  ConfigurableNavigationHandler nav = (ConfigurableNavigationHandler) fc  .getApplication().getNavigationHandler();  addMessage("must.authentify", Constants.***ERROR\_MSG***);  nav.performNavigation("/authentication/login");  **return**;  }  **if** (!userVo.haseRole(Constants.***ADMIN\_ROLE***)) {  ConfigurableNavigationHandler nav = (ConfigurableNavigationHandler) fc  .getApplication().getNavigationHandler();  nav.performNavigation("accessDenied");  }  }  **public** **void** isClient(ComponentSystemEvent event) {  FacesContext fc = FacesContext.*getCurrentInstance*();  **if** (Utils.*getSession*().getAttribute("user") == **null**) {  ConfigurableNavigationHandler nav = (ConfigurableNavigationHandler) fc  .getApplication().getNavigationHandler();  addMessage("must.authentify", Constants.***ERROR\_MSG***);  nav.performNavigation("/authentication/login");  **return**;  }  **if** (!userVo.haseRole(Constants.***CLIENT\_ROLE***)) {  ConfigurableNavigationHandler nav = (ConfigurableNavigationHandler) fc  .getApplication().getNavigationHandler();  nav.performNavigation("accessDenied");  }  }  } |

|  |
| --- |
| **La classe LangueBean :** |
| **package** ma.gov.formation.cigma.presentation.internationalization;  **import** java.io.Serializable;  **import** java.util.Locale;  **import** javax.faces.context.FacesContext;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.presentation.BaseAction;  **import** org.springframework.context.annotation.Scope;  @Named("langue")  @Scope("session")  **public** **class** LangueBean **extends** BaseAction **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **private** Locale locale = **new** Locale("en");  **public** Locale getLocale() {  **return** locale;  }  **public** String getLanguage() {  **return** locale.getLanguage();  }  **public** **void** changeLanguage(String language) {  locale = **new** Locale(language);  FacesContext.*getCurrentInstance*().getViewRoot().setLocale(locale);  }  } |

|  |
| --- |
| **La classe PanierBean :** |
| **package** ma.gov.formation.cigma.presentation.panier;  **import** java.util.ArrayList;  **import** java.util.List;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.presentation.BaseAction;  **import** org.springframework.context.annotation.Scope;  @Named("panierBean")  @Scope("session")  **public** **class** PanierBean **extends** BaseAction {  **private** **static** List<String> *images*;  **static** {  *images* = **new** ArrayList<String>();  **for** (**int** i = 1; i <= 6; i++) {  *images*.add("/images/articles/photo" + i + ".jpg");  }  }  **public** List<String> getImages() {  **return** *images*;  }  **public** String list() {  **return** "/panier/panier.xhtml";  }  } |

|  |
| --- |
| **La classe ThemeBean :** |
| **package** ma.gov.formation.cigma.presentation.theme;  **import** java.io.Serializable;  **import** java.util.List;  **import** javax.annotation.PostConstruct;  **import** javax.inject.Inject;  **import** javax.inject.Named;  **import** ma.gov.formation.cigma.domaine.theme.ThemeVo;  **import** ma.gov.formation.cigma.presentation.BaseAction;  **import** ma.gov.formation.cigma.presentation.Constants;  **import** ma.gov.formation.cigma.service.IThemeService;  **import** org.springframework.context.annotation.Scope;  @Named("themeBean")  @Scope("application")  **public** **class** ThemeBean **extends** BaseAction **implements** Serializable{  **private** **static** **final** **long** ***serialVersionUID*** = 2510779045682332364L;  **private** List<ThemeVo> themes;  **private** String selectedTheme;  @Inject  IThemeService themeService;  **public** String getSelectedTheme() {  **return** selectedTheme;  }  **public** **void** setSelectedTheme(String selectedTheme) {  **this**.selectedTheme = selectedTheme;  }  @PostConstruct  **public** **void** init() {  **try** {  setSelectedTheme(themeService.getDefaultTheme().getCode());  themes = themeService.getAll(**new** ThemeVo());  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  }    }  **public** **void** setThemes(List<ThemeVo> themes) {  **this**.themes = themes;  }  **public** List<ThemeVo> getThemes() {  **return** themes;  }  **public** String edit() {  **return** "/admin/theme.xhtml";  }  **public** **void** saveTheme(ThemeVo theme) {  **try** {  themeService.updateThemeByDefault(selectedTheme);  setSelectedTheme(selectedTheme);  } **catch** (Exception e) {  addMessage("probleme.technique", Constants.***ERROR\_MSG***);  e.printStackTrace();  }  }  } |

1. **Les fichiers \*.xhtml**

|  |
| --- |
| **Créer la page WebContent/mainTemplate/mainTemplate.xml** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head />  <h:body>  <p:layout fullPage=*"true"*>  <f:view locale=*"#{langue.locale}"*>  <p:layoutUnit position=*"north"* size=*"60"*>  <ui:include src=*"/mainTemplate/commonHeader.xhtml"* />  </p:layoutUnit>  <p:layoutUnit position=*"west"* size=*"250"* resizable=*"true"*  closable=*"true"* collapsible=*"true"*>  <ui:include src=*"/mainTemplate/commonMenu.xhtml"* />  </p:layoutUnit>  <p:layoutUnit position=*"center"*>  <h:form id=*"templatePrincipal"*>  <p:growl id=*"msgs"* showDetail=*"true"*/>  </h:form>  <ui:insert name=*"body"*>  <ui:include src=*"/mainTemplate/commonBody.xhtml"* />  </ui:insert>  </p:layoutUnit>  <p:layoutUnit position=*"east"* size=*"320"* resizable=*"true"*>  <p:calendar id=*"inline"* value=*""* mode=*"inline"* />  </p:layoutUnit>  <p:layoutUnit position=*"south"* size=*"80"*>  <ui:include src=*"/mainTemplate/commonFooter.xhtml"* />  </p:layoutUnit>  </f:view>  </p:layout>  </h:body>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/mainTemplate/commonHeader.xml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head />  <ui:composition>  <h:form>  <p:menuButton value=*"#{msg.preference}"*>  <p:menuitem value=*"#{msg.profile}"* action=*"#{loginBean.toDo}"* ajax=*"false"*/>  <p:menuitem value=*"#{msg.updatepwd}"* action=*"#{loginBean.toDo}"* ajax=*"false"*/>  <p:separator/>  <p:menuitem value=*"#{msg.logout}"* action=*"#{loginBean.logout}"* ajax=*"false"*/>  </p:menuButton>  Connected User : <h:outputLabel value=*"#{user.userName}"*/>  <h:commandLink action=*"#{langue.changeLanguage('ar')}"* style="float:*right*">  <h:graphicImage value=*"/resources/images/internationalization/ar.png"* />  </h:commandLink>  <h:commandLink action=*"#{langue.changeLanguage('fr')}"* style="float:*right*">  <h:graphicImage value=*"/resources/images/internationalization/fr.png"* />  </h:commandLink>  <h:commandLink action=*"#{langue.changeLanguage('en')}"* style="float:*right*">  <h:graphicImage value=*"/resources/images/internationalization/en.png"* />  </h:commandLink>  <br></br>      </h:form>  </ui:composition>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/mainTemplate/commonMenu.xml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head/>  <h:body>  <ui:composition>  <h:form>  <p:panelMenu style="width:*300px*">  <p:submenu label=*"#{msg['menu.admin']}"* rendered=*"#{user.admin}"*>  <p:menuitem value=*"#{msg['menu.admin.sm.gestArt']}"* action=*"#{articleBean.list}"* ajax=*"false"*/>  <p:submenu label=*"#{msg['menu.admin.sm.gestUtil']}"* icon=*"pi pi-home"*>  <p:menuitem value=*"#{msg['menu.admin.sm.gestUtil.sm.addUser']}"* action=*"#{articleBean.toDo}"* ajax=*"false"*/>  <p:menuitem value=*"#{msg['menu.admin.sm.gestUtil.sm.addRole']}"* action=*"#{articleBean.toDo}"* ajax=*"false"*/>  <p:menuitem value=*"#{msg['menu.admin.sm.gestUtil.sm.listUsers']}"* action=*"#{articleBean.toDo}"* ajax=*"false"*/>  </p:submenu>  </p:submenu>  <p:separator />  <p:submenu label=*"#{msg['menu.preference']}"* rendered=*"#{user.admin}"*>  <p:menuitem value=*"#{msg['menu.preference.sm.Theme']}"* action=*"#{themeBean.edit}"* ajax=*"false"*/>  </p:submenu>  <p:separator />  <p:submenu label=*"#{msg['menu.panier']}"* rendered=*"#{user.client}"*>  <p:menuitem value=*"#{msg['menu.panier.sb.consulter']}"* action=*"#{panierBean.list}"* ajax=*"false"*/>  <p:menuitem value=*"#{msg['menu.panier.sb.edit']}"* action=*"#{panierBean.toDo}"* ajax=*"false"*/>  <p:menuitem value=*"#{msg['menu.panier.sb.acheter']}"* action=*"#{panierBean.toDo}"* ajax=*"false"*/>  <p:menuitem value=*"#{msg['menu.panier.sb.editerfact']}"* action=*"#{panierBean.toDo}"* ajax=*"false"*/>  </p:submenu>  </p:panelMenu>  </h:form>  </ui:composition>  </h:body>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/mainTemplate/commonFooter.xml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head/>  <h:body>  <ui:composition>  <h:form>  <p:dock position=*"bottom"*>  <p:menuitem value=*"Home"* icon=*"/resources/images/footer/home.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"Music"* icon=*"/resources/images/footer/music.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"Video"* icon=*"/resources/images/footer/video.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"Email"* icon=*"/resources/images/footer/email.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"Portfolio"* icon=*"/resources/images/footer/portfolio.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"Link"* icon=*"/resources/images/footer/link.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"RSS"* icon=*"/resources/images/footer/rss.png"* action=*"#{loginBean.toDo}"*/>  <p:menuitem value=*"History"* icon=*"/resources/images/footer/history.png"* action=*"#{loginBean.toDo}"*/>  </p:dock>  </h:form>  </ui:composition>  </h:body>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/mainTemplate/commonBody.xml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head/>  <h:body>  <ui:composition>  Contenu à mettre ici  </ui:composition>  </h:body>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/admin/article/articles.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head/>  <ui:composition template=*"/mainTemplate/mainTemplate.xhtml"*>  <ui:define name=*"body"*>  <ui:include src=*"/admin/article/ajouter.xhtml"* />  <ui:include src=*"/admin/article/liste.xhtml"* />  </ui:define>  </ui:composition>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/admin/article/ajouter.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <ui:composition>  <h:form enctype=*"multipart/form-data"*>  <p:panelGrid columns=*"2"*>  <f:facet name=*"header"*>  <h:outputText value=*"#{msg['ajouter.nouvel.article']}"* />  </f:facet>  <h:outputLabel value=*"#{msg['formulaire.article.designation']}"* />  <h:inputText id=*"designation"*  value=*"#{articleBean.articleVo.designation}"* required=*"true"*  immediate=*"true"*>  </h:inputText>  <h:outputLabel value=*"#{msg['formulaire.article.prix']}"* />  <h:inputText id=*"price"* value=*"#{articleBean.articleVo.prix}"* />  <h:outputLabel value=*"#{msg['formulaire.article.quantiteEnStock']}"* />  <h:inputText id=*"quantiteEnStock"* value=*"#{articleBean.articleVo.quantiteEnStock}"* />    <h:outputLabel value=*"Catégorie"* />  <h:selectOneMenu value=*"#{articleBean.articleVo.categorie.code}"* style="width:*100%*">  <f:selectItems value=*"#{articleBean.categories}"* var=*"categorie"* itemLabel=*"#{categorie.libelle}"* itemValue=*"#{categorie.code}"*/>  </h:selectOneMenu>    <h:outputLabel value=*"Photo"* />  <p:fileUpload value=*"#{articleBean.file}"* mode=*"simple"* skinSimple=*"true"* allowTypes=*"/(\.|\/)(gif|jpe?g|png)$/"*/>  <f:facet name=*"footer"*>  <h:panelGroup style="display:*block*; text-align:*center*">  <p:commandButton value=*"#{msg['formulaire.bouton.ajouter']}"* ajax=*"false"* action=*"#{articleBean.addArticle}"*/>  </h:panelGroup>  </f:facet>  </p:panelGrid>  </h:form>  </ui:composition>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/admin/article/liste.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head/>  <ui:composition>  <h:form>  <h2>#{msg['message.listeDesArticles']}</h2>  <p:dataTable id=*"tbl"* var=*"article"* value=*"#{articleBean.articlesVo}"* editable=*"true"* paginatorTemplate=*"{CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink} {Exporters}"*  paginator=*"true"* rows=*"5"* style="margin-bottom:*20px*">  <f:facet name=*"{Exporters}"*>  <div style="float: *right*; padding-top: *5px*;">  <h:commandLink>  <p:graphicImage name=*"images/export/excel.png"* width=*"24"* />  <p:dataExporter type=*"xls"* target=*"tbl"* fileName=*"#{msg.file\_name}"* />  </h:commandLink>  <h:commandLink>  <p:graphicImage name=*"images/export/pdf.png"* width=*"24"* />  <p:dataExporter type=*"pdf"* target=*"tbl"* fileName=*"#{msg.file\_name}"*/>  </h:commandLink>  <h:commandLink>  <p:graphicImage name=*"images/export/csv.png"* width=*"24"* />  <p:dataExporter type=*"csv"* target=*"tbl"* fileName=*"#{msg.file\_name}"* />  </h:commandLink>  <h:commandLink>  <p:graphicImage name=*"images/export/xml.png"* width=*"24"* />  <p:dataExporter type=*"xml"* target=*"tbl"* fileName=*"#{msg.file\_name}"* />  </h:commandLink>  </div>  </f:facet>  <p:ajax event=*"rowEdit"* listener=*"#{articleBean.onRowEdit}"* update=*":templatePrincipal:msgs"* />  <p:ajax event=*"rowEditCancel"* listener=*"#{articleBean.onRowCancel}"* update=*":templatePrincipal:msgs"* />      <p:column headerText=*"#{msg['intitule.colonne.identifiant.listeDesArticles']}"*>  <p:cellEditor>  <f:facet name=*"output"*><h:outputText value=*"#{article.id}"* /></f:facet>  <f:facet name=*"input"*><p:inputText id=*"modelInputid"* value=*"#{article.id}"* style="width:*100%*"/></f:facet>  </p:cellEditor>  </p:column>    <p:column headerText=*"#{msg['intitule.colonne.designation.listeDesArticles']}"*>  <p:cellEditor>  <f:facet name=*"output"*><h:outputText value=*"#{article.designation}"* /></f:facet>  <f:facet name=*"input"*><p:inputText id=*"modelInputdesc"* value=*"#{article.designation}"* style="width:*100%*"/></f:facet>  </p:cellEditor>  </p:column>    <p:column headerText=*"#{msg['intitule.colonne.prix.listeDesArticles']}"*>  <p:cellEditor>  <f:facet name=*"output"*><h:outputText value=*"#{article.prix}"* /></f:facet>  <f:facet name=*"input"*><p:inputText id=*"modelInputprix"* value=*"#{article.prix}"* style="width:*100%*"/></f:facet>  </p:cellEditor>  </p:column>      <p:column headerText=*"#{msg['intitule.colonne.quantiteEnStock.listeDesArticles']}"*>  <p:cellEditor>  <f:facet name=*"output"*><h:outputText value=*"#{article.quantiteEnStock}"* /></f:facet>  <f:facet name=*"input"*><p:inputText id=*"modelInputquan"* value=*"#{article.quantiteEnStock}"* style="width:*100%*"/></f:facet>  </p:cellEditor>  </p:column>    <p:column headerText=*"Catgorie"*>  <p:cellEditor>  <f:facet name=*"output"*><h:outputText value=*"#{article.categorie.libelle}"* /></f:facet>  <f:facet name=*"input"*>  <h:selectOneMenu value=*"#{article.categorie.code}"* style="width:*100%*">  <f:selectItems value=*"#{articleBean.categories}"* var=*"categorie"* itemLabel=*"#{categorie.code}"* itemValue=*"#{categorie.libelle}"* />  </h:selectOneMenu>  </f:facet>  </p:cellEditor>  </p:column>  <p:column style="width:*35px*" exportable=*"false"*>  <p:rowEditor />  </p:column>  </p:dataTable>  </h:form>  </ui:composition>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/admin/theme.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <h:head />  <ui:composition template=*"/mainTemplate/mainTemplate.xhtml"*>  <ui:define name=*"body"*>  <h:form>  <h:panelGrid columns=*"2"* cellpadding=*"10"*>  <p:outputLabel for=*"basic"* value=*"Basic:"* />  <p:themeSwitcher id=*"basic"* style="width:*165px*" value=*"#{themeBean.selectedTheme}"*>  <f:selectItem itemLabel=*"#{msg['choose.theme']}"* itemValue=*""* />  <f:selectItems value=*"#{themeBean.themes}"* var=*"theme"* itemLabel=*"#{theme.libelle}"* itemValue=*"#{theme.code}"*/>  <f:ajax listener=*"#{themeBean.saveTheme(theme)}"*></f:ajax>  </p:themeSwitcher>  </h:panelGrid>  </h:form>  </ui:define>  </ui:composition>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/** **authentication/login.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head />  <h:body>  <h:form id=*"LoginForm"*>  <p:growl id=*"msgs"* showDetail=*"true"* />  <p:panelGrid columns=*"2"*>  <f:facet name=*"header"*>  <h:outputText value=*"#{msg['se.connecter']}"* />  </f:facet>  <p:outputLabel value=*"#{msg['formulaire.login.username']}"* />  <p:inputText id=*"userName"* value=*"#{loginBean.userVo.userName}"* size=*"40"* required=*"true"* />  <p:outputLabel value=*"#{msg['formulaire.login.password']}"* />  <p:password id=*"password"* value=*"#{loginBean.userVo.password}"* size=*"40"* required=*"true"* />  <f:facet name=*"footer"*>  <h:panelGroup>  <p:commandButton value=*"#{msg['login.bouton.sautentifier']}"* action=*"#{loginBean.login}"* update=*":LoginForm:msgs"* />  </h:panelGroup>  </f:facet>  </p:panelGrid>  </h:form>  </h:body>  </f:view>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/panier/panier.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head />  <ui:composition template=*"/mainTemplate/mainTemplate.xhtml"*>  <ui:define name=*"body"*>  <h:form>  <h3>Liste des aticles par catégories :</h3>  <p:accordionPanel multiple=*"true"*>  <p:tab title=*"Matetriels informatiques"*>  <p:galleria value=*"#{panierBean.images}"* var=*"image"*  panelWidth=*"600"* panelHeight=*"200"* showCaption=*"true"*>  <p:graphicImage name=*"#{image}"*  alt=*"Image Description for #{image}"* title=*"#{image}"* />  </p:galleria>  </p:tab>  <p:tab title=*"Sport"*>  <p:galleria value=*"#{panierBean.images}"* var=*"image"*  panelWidth=*"600"* panelHeight=*"200"* showCaption=*"true"*>  <p:graphicImage name=*"#{image}"*  alt=*"Image Description for #{image}"* title=*"#{image}"* />  </p:galleria>  </p:tab>  <p:tab title=*"Autre"*>  <p:galleria value=*"#{panierBean.images}"* var=*"image"*  panelWidth=*"600"* panelHeight=*"200"* showCaption=*"true"*>  <p:graphicImage name=*"#{image}"*  alt=*"Image Description for #{image}"* title=*"#{image}"* />  </p:galleria>  </p:tab>  </p:accordionPanel>  </h:form>  </ui:define>  </ui:composition>  </f:view>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/accessDenied.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head />  <ui:composition template=*"/mainTemplate/mainTemplate.xhtml"*>  <ui:define name=*"body"*>  <h:form>  <h3>  <h:outputFormat value=*"#{msg['message.access.denied']}"*>  <f:param value=*"#{user.userName}"*></f:param></h:outputFormat>  </h3>  </h:form>  </ui:define>  </ui:composition>  </f:view>  </h:html> |

|  |
| --- |
| **Créer la page WebContent/toDo.xhtml :** |
| <h:html lang=*"#{langue.language}"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://xmlns.jcp.org/jsf/core"*  xmlns:ui=*"http://xmlns.jcp.org/jsf/facelets"*  xmlns:a=*"http://xmlns.jcp.org/jsf/passthrough"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head />  <ui:composition template=*"/mainTemplate/mainTemplate.xhtml"*>  <ui:define name=*"body"*>  <h:form>  <h3>To be completed</h3>  </h:form>  </ui:define>  </ui:composition>  </f:view>  </h:html> |

1. **Le fichier messages.properties**

|  |
| --- |
| **src/messages.properties :** |
| #page articles.xhtml et edit.xhtml  titre=Web Application with : JSF, JPA and Wildfly  formulaire.article.designation=Article Designation  formulaire.article.prix=Price  formulaire.article.quantiteEnStock=Quantity in stock  #page articles.xhtml  ajouter.nouvel.article=Add a new article :  formulaire.bouton.ajouter=Add  message.listeDesArticles=Articles list :  intitule.colonne.identifiant.listeDesArticles=Identifier  intitule.colonne.designation.listeDesArticles=Designation  intitule.colonne.prix.listeDesArticles=Price  intitule.colonne.quantiteEnStock.listeDesArticles=Quantity in stock  intitule.lien.supprimer=Delete  intitule.lien.modifier=Update  #page edit.xhtml  modifier.article=Update an article :  formulaire.bouton.modifier=Update  #page login.xhtml  se.connecter=Autentication :  formulaire.login.username=Login  formulaire.login.password=Password  login.bouton.sautentifier=Login  logout=Logout  #page panier.xhtml  msg.panier=Panier  #page accessDenied.xhtml  message.access.denied=Hello **{0}** : You are not allowed to access this page. Please contact the administrator.  lien.revenir.page.login=Go Back to authentication page  #Page menu :  menu.admin=Administration  menu.admin.sm.gestArt=Article Management  menu.admin.sm.gestUtil=User Management  menu.admin.sm.gestUtil.sm.addUser=Create User  menu.admin.sm.gestUtil.sm.addRole=Create Role  menu.admin.sm.gestUtil.sm.listUsers=Users List  menu.admin.sm.parametres=Parameters  menu.panier=Basket Management  menu.panier.sb.consulter=Consult Articles  menu.panier.sb.edit=Consulter panier  menu.panier.sb.acheter=Buy Article  menu.panier.sb.editerfact=Edit Invoice  menu.preference=Preference  menu.preference.sm.Theme=Update Theme  footer=JSF training: use of the facelets  logout=Logout  # Page theme :  choose.theme=Choose Theme  #Validations messages  login.doesnt.exist=Login doesnt exist  password.incorrect=Password incorrect  problem.user.role=You dont have any role. Please contact your administrator  save.update=Enregistrer les modifications  cancel.update=Annuler les modifications  must.authentify=Merci de vous vous authentifiez pour accéder à l'application.  #Infos messages  article.ajoute.avec.succes=Article added successfully.  article.supprime.avec.succes=article successfully deleted.  article.modifier.avec.succes=Articles successfully edited.  modifications.annulees=Modifications canceled.  probleme.copie.fichier=Problem of uploading the photo of the article.  #Page header  preference=Preference  profile=Profile  updatepwd=Update password  probleme.technique=Technical problem.  file\_name=Liste\_Articles |

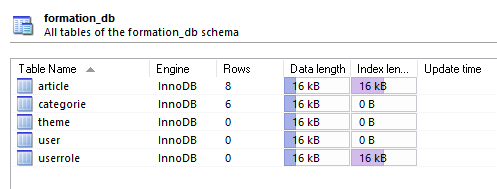
1. **Les images**

Copier le dossier images dans le répertoire WebContent/resources.

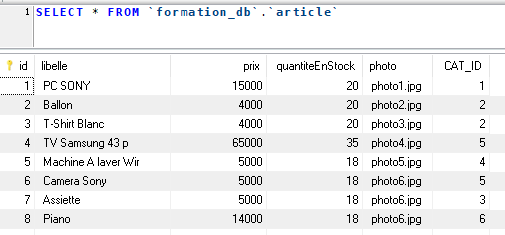
1. **Initialisation de la base de données :**

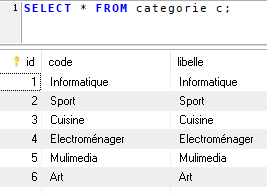
|  |
| --- |
| Créer la classe test.article.Test et exécuter la méthode main ci-dessous afin d'insérer des données au niveau des deux tables CATEGORIE et ARTICLE.  Remplacer le chemin du fichier applicationContext.xml par le votre. |
| **package** test.article;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  **import** ma.gov.formation.cigma.domaine.categorie.CategorieVo;  **import** ma.gov.formation.cigma.service.IArticleService;  **import** ma.gov.formation.cigma.service.IBaseService;  **import** org.springframework.context.ApplicationContext;  **import** org.springframework.context.support.FileSystemXmlApplicationContext;  **public** **class** Test {  **public** **static** **void** main(String[] args) {  **try** {  ApplicationContext context = **new** FileSystemXmlApplicationContext(  **new** String[] { "C:/workspace/JSF/tp4/WebContent/WEB-INF/applicationContext.xml" });  IArticleService articleService = (IArticleService) context  .getBean("articleService");    IBaseService baseService = (IBaseService) context  .getBean("baseService");  CategorieVo c1 = **new** CategorieVo("Informatique", "Informatique");  CategorieVo c2 = **new** CategorieVo("Sport", "Sport");  CategorieVo c3 = **new** CategorieVo("Cuisine", "Cuisine");  CategorieVo c4 = **new** CategorieVo("Electroménager", "Electroménager");  CategorieVo c5 = **new** CategorieVo("Mulimedia", "Mulimedia");  CategorieVo c6 = **new** CategorieVo("Art", "Art");  ArticleVo a1 = **new** ArticleVo("PC SONY", 15000.0, 20, "photo1.jpg");  ArticleVo a2 = **new** ArticleVo("Ballon", 4000.0, 20, "photo2.jpg");  ArticleVo a3 = **new** ArticleVo("T-Shirt Blanc", 4000.0, 20,"photo3.jpg");  ArticleVo a4 = **new** ArticleVo("TV Samsung 43 p", 65000.0, 35,"photo4.jpg");  ArticleVo a5 = **new** ArticleVo("Machine A laver Wir", 5000.0, 18, "photo5.jpg");  ArticleVo a6 = **new** ArticleVo("Camera Sony", 5000.0, 18,"photo6.jpg");  ArticleVo a7 = **new** ArticleVo("Assiette", 5000.0, 18, "photo6.jpg");  ArticleVo a8 = **new** ArticleVo("Piano", 14000.0, 18, "photo6.jpg");  baseService.save(c1);  baseService.save(c2);  baseService.save(c3);  baseService.save(c4);  baseService.save(c5);  baseService.save(c6);  a1.setCategorie(c1);  a2.setCategorie(c2);  a3.setCategorie(c2);  a4.setCategorie(c5);  a6.setCategorie(c5);  a5.setCategorie(c4);  a7.setCategorie(c3);  a8.setCategorie(c6);    articleService.save(a1);  articleService.save(a2);  articleService.save(a3);  articleService.save(a4);  articleService.save(a5);  articleService.save(a6);  articleService.save(a7);  articleService.save(a8);    } **catch** (Exception e) {  e.printStackTrace();  }  }  } |

Vérifier que les 05 tables suivantes ont été crées automatiquement :

****

Vérifier également que les données de test ont été insérées au niveau des deux tables CATEGORIE et ARTICLE.

****

****

|  |
| --- |
| **Créer la classe test.theme.Test et exécuter la méthode main ci-dessous :**  Remplacer le chemin du fichier applicationContext.xml par le votre. |
| **package** test.theme;  **import** ma.gov.formation.cigma.domaine.theme.ThemeVo;  **import** ma.gov.formation.cigma.service.IThemeService;  **import** org.springframework.context.ApplicationContext;  **import** org.springframework.context.support.FileSystemXmlApplicationContext;  **public** **class** Test {  **public** **static** **void** main(String[] args) {  **try** {  ApplicationContext context = **new** FileSystemXmlApplicationContext(  **new** String[] { "C:/workspace/JSF/tp4/WebContent/WEB-INF/applicationContext.xml" });  IThemeService service = (IThemeService) context  .getBean("themeService");  ThemeVo theme1 = **new** ThemeVo("glass-x", "glass-x");  ThemeVo theme2 = **new** ThemeVo("cruze", "cruze");  ThemeVo theme3 = **new** ThemeVo("blitzer", "blitzer");  ThemeVo theme4 = **new** ThemeVo("afternoon", "afternoon", "YES");  ThemeVo theme5 = **new** ThemeVo("afterdark", "afterdark");  ThemeVo theme6 = **new** ThemeVo("bootstrap", "bootstrap");  ThemeVo theme7 = **new** ThemeVo("bluesky", "bluesky");  ThemeVo theme8 = **new** ThemeVo("midnight", "midnight");  ThemeVo theme9 = **new** ThemeVo("cupertino", "cupertino");  ThemeVo theme10 = **new** ThemeVo("aristo", "aristo");  ThemeVo theme11 = **new** ThemeVo("blitzer", "blitzer");  ThemeVo theme12 = **new** ThemeVo("casablanca", "casablanca");  ThemeVo theme13 = **new** ThemeVo("dark-hive", "dark-hive");  ThemeVo theme14 = **new** ThemeVo("delta", "delta");  ThemeVo theme15 = **new** ThemeVo("dot-luv", "dot-luv");  service.save(theme1);  service.save(theme2);  service.save(theme3);  service.save(theme4);  service.save(theme5);  service.save(theme6);  service.save(theme7);  service.save(theme8);  service.save(theme9);  service.save(theme10);  service.save(theme11);  service.save(theme12);  service.save(theme13);  service.save(theme14);  service.save(theme15);  } **catch** (Exception e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

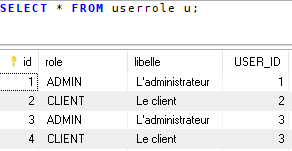
Vérfier que les données ont été insérées au niveau de la table THEME :

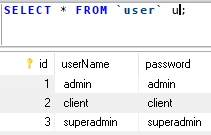


Remarquez que le thème par défaut qui est utilisé par l'application est **"afternoon".**

|  |
| --- |
| Créer la classe test.user.Test et exécuter la méthode main ci-dessous :  Remplacer le chemin du fichier applicationContext.xml par le votre. |
| **package** test.user;  **import** ma.gov.formation.cigma.domaine.user.UserRoleVo;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **import** ma.gov.formation.cigma.presentation.Constants;  **import** ma.gov.formation.cigma.service.IUserService;  **import** org.springframework.context.ApplicationContext;  **import** org.springframework.context.support.FileSystemXmlApplicationContext;  **public** **class** Test {  **public** **static** **void** main(String[] args) {  **try** {  ApplicationContext context = **new** FileSystemXmlApplicationContext(  **new** String[] { "C:/workspace/JSF/tp4/WebContent/WEB-INF/applicationContext.xml" });  IUserService userService = (IUserService) context  .getBean("userService");    UserRoleVo adminRole1=**new** UserRoleVo(Constants.***ADMIN\_ROLE***,Constants.***ADMIN\_ROLE\_DESCRIPTION***);  UserRoleVo adminRole2=**new** UserRoleVo(Constants.***ADMIN\_ROLE***,Constants.***ADMIN\_ROLE\_DESCRIPTION***);  UserRoleVo clientRole1=**new** UserRoleVo(Constants.***CLIENT\_ROLE***,Constants.***CLIENT\_ROLE\_DESCRIPTION***);  UserRoleVo clientRole2=**new** UserRoleVo(Constants.***CLIENT\_ROLE***,Constants.***CLIENT\_ROLE\_DESCRIPTION***);    UserVo user1=**new** UserVo("admin","admin");user1.getRoles().add(adminRole1);  UserVo user2=**new** UserVo("client","client");user2.getRoles().add(clientRole1);  UserVo user3=**new** UserVo("superadmin","superadmin");user3.getRoles().add(adminRole2);user3.getRoles().add(clientRole2);  userService.save(user1);  userService.save(user2);  userService.save(user3);  } **catch** (Exception e) {  e.printStackTrace();  }  }  } |

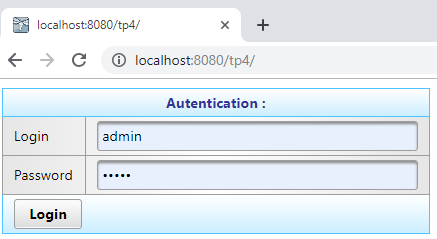
Vérfier que les données suivantes ont été bien insérées :



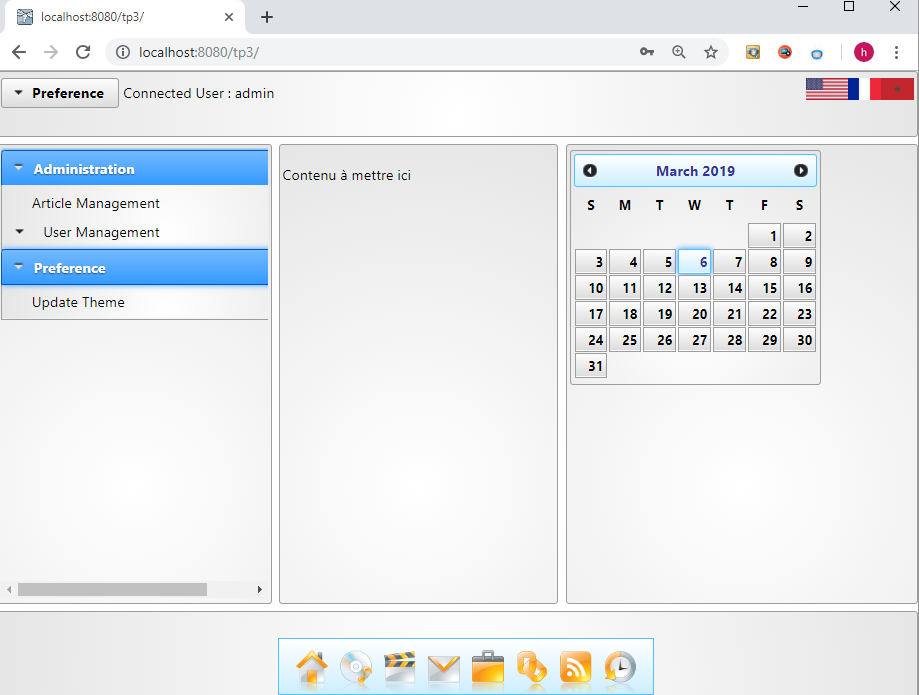


1. **Test de l'application**

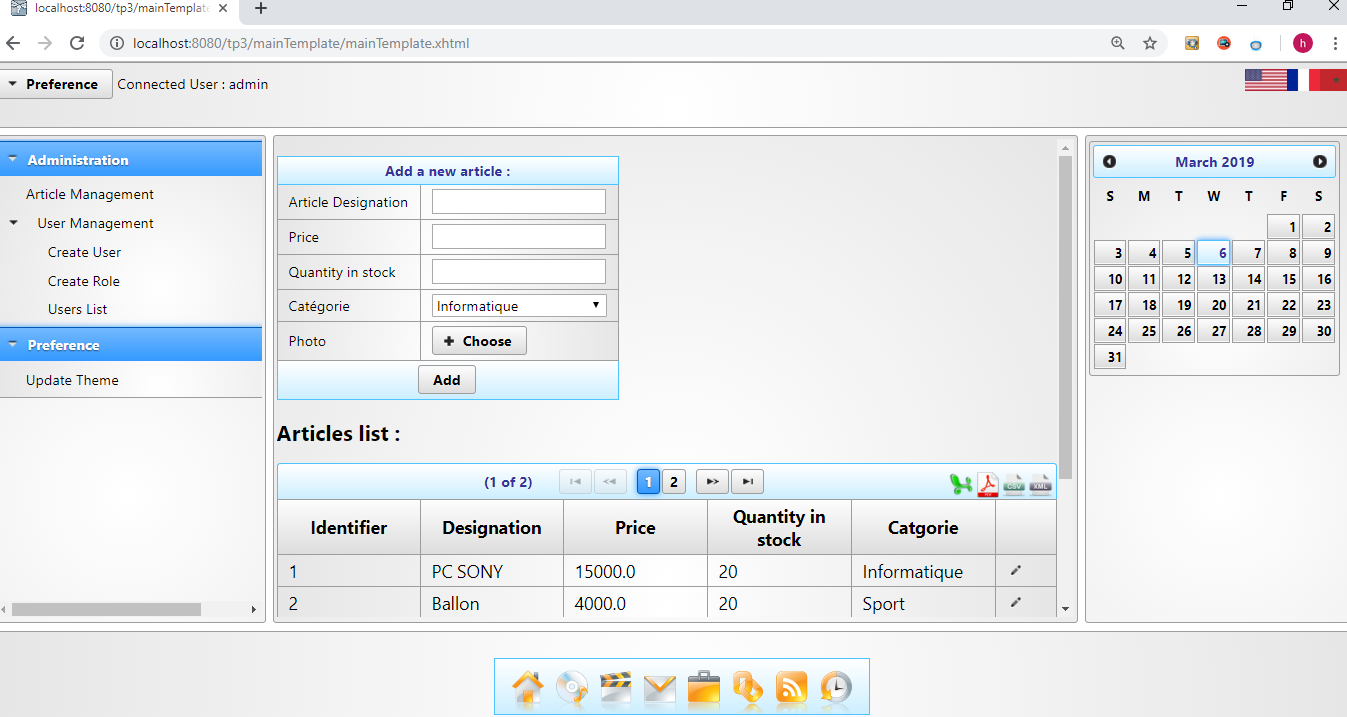
Déployer votre application au niveau de Wildfly et ensuite lancer le lien **http://localhost:8080/tp4.** La page suivante sera affichée :

****

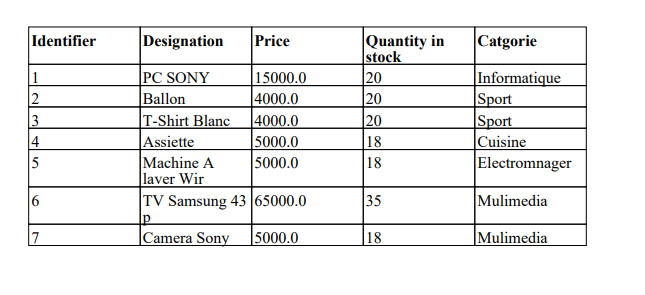
Entrer un compte utilisateur (par exemple : admin/admin). La page suivante sera affichée :



\* Pour la gestion des articles, cliquer sur le menu "Article Management" ci-dessus :

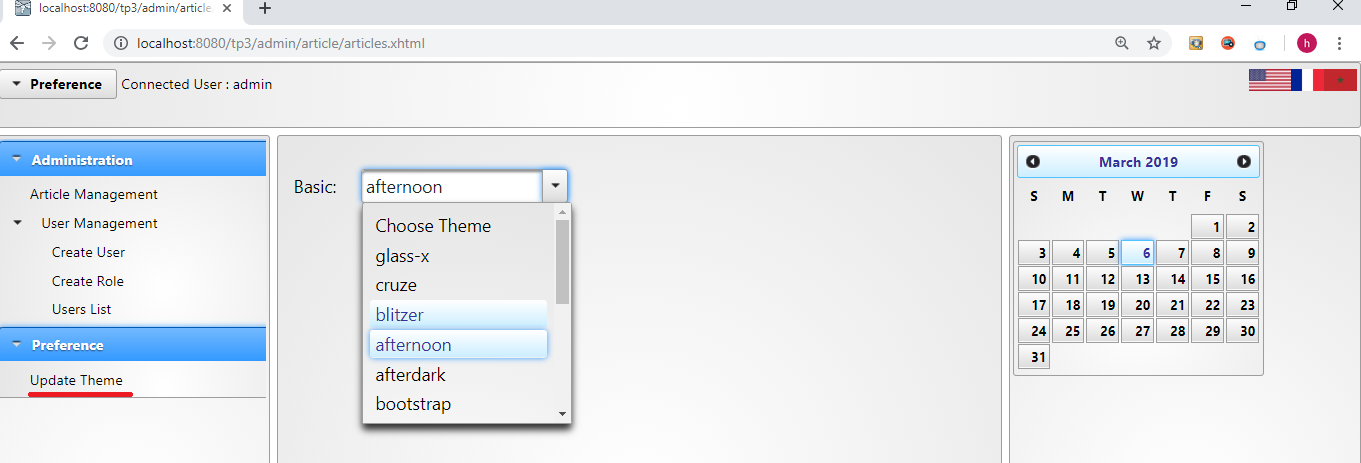


\* Pour exporter les articles en fichier PDF, cliquer sur l'icon "pdf" et vérifier que le fichier List\_Articles.pdf a été bien généré :

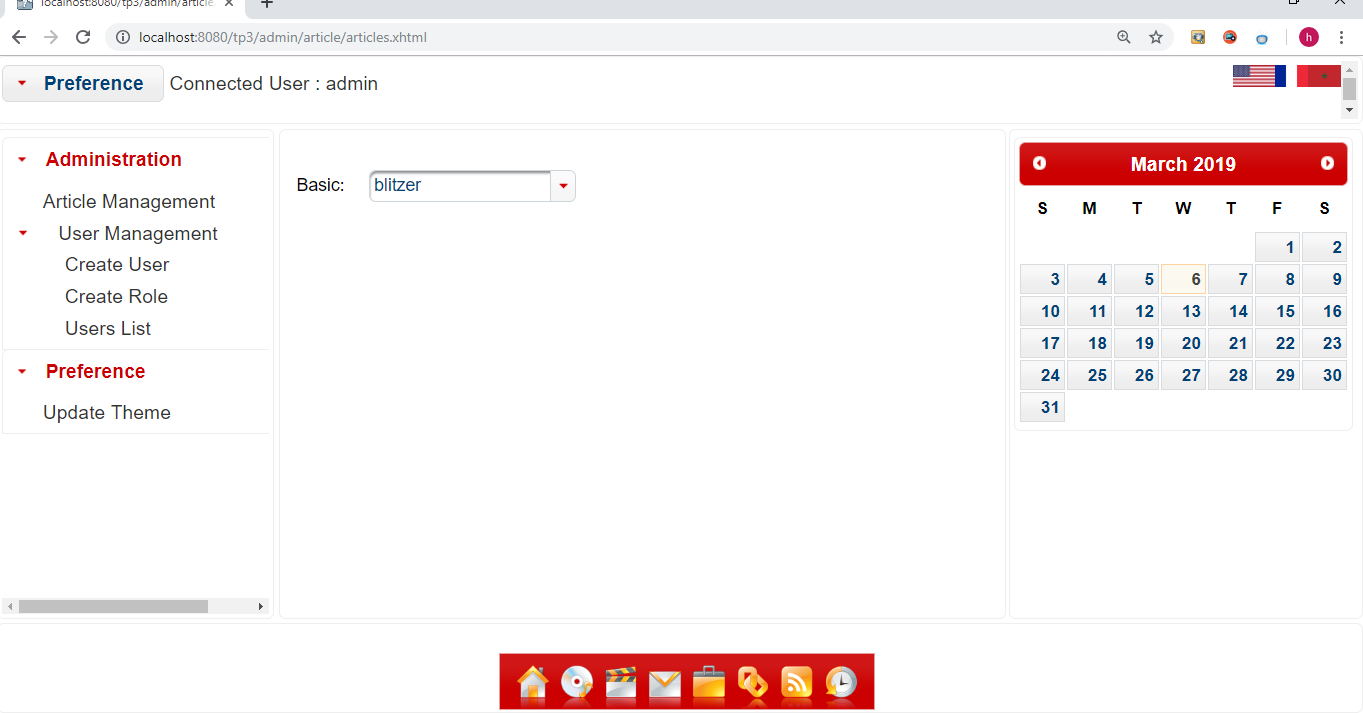


\* Idem pour les autres formats : Excel, XML et CSV.

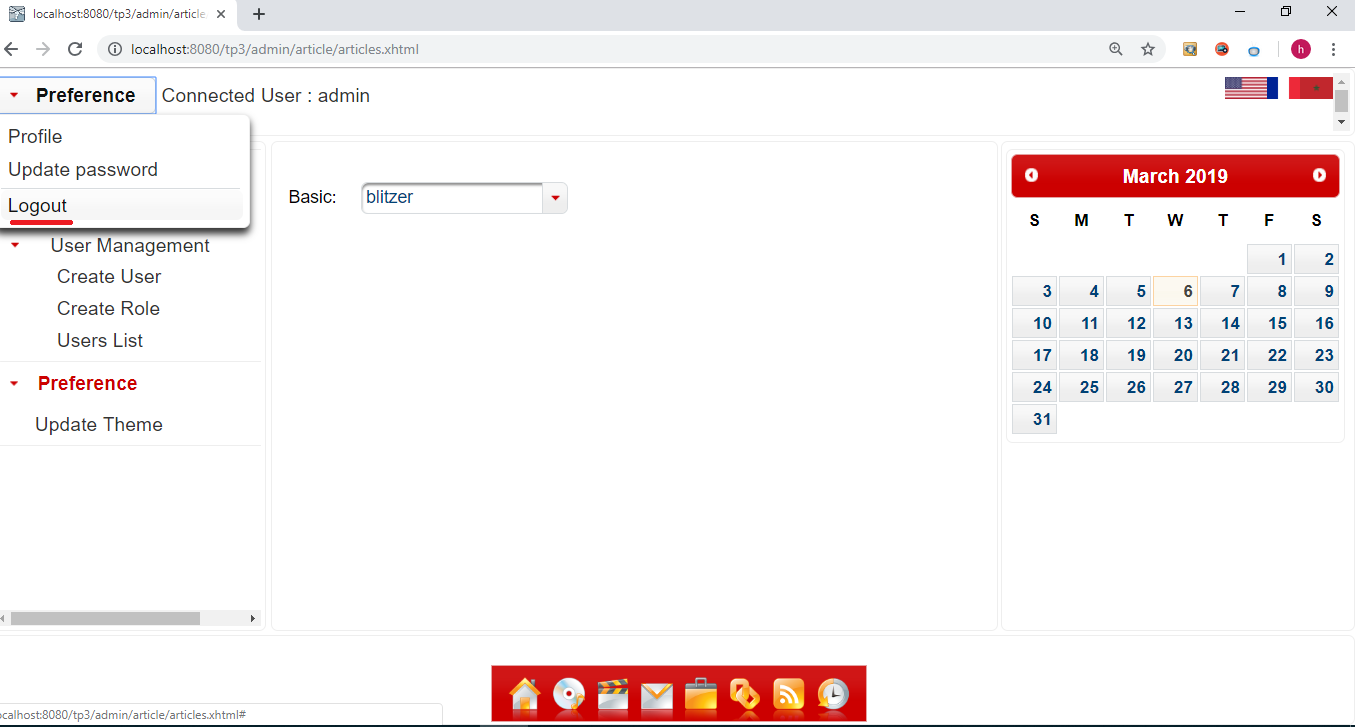
\* Pour changer le thème, cliquer sur "Update Theme" comme le montre l'écran suivant :



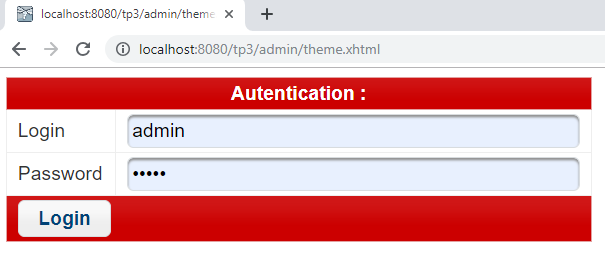
\* Choisir un autre thème, par exemple "blitzer". Vérifier que le thème a été appliqué à l'ensemble de vos pages :



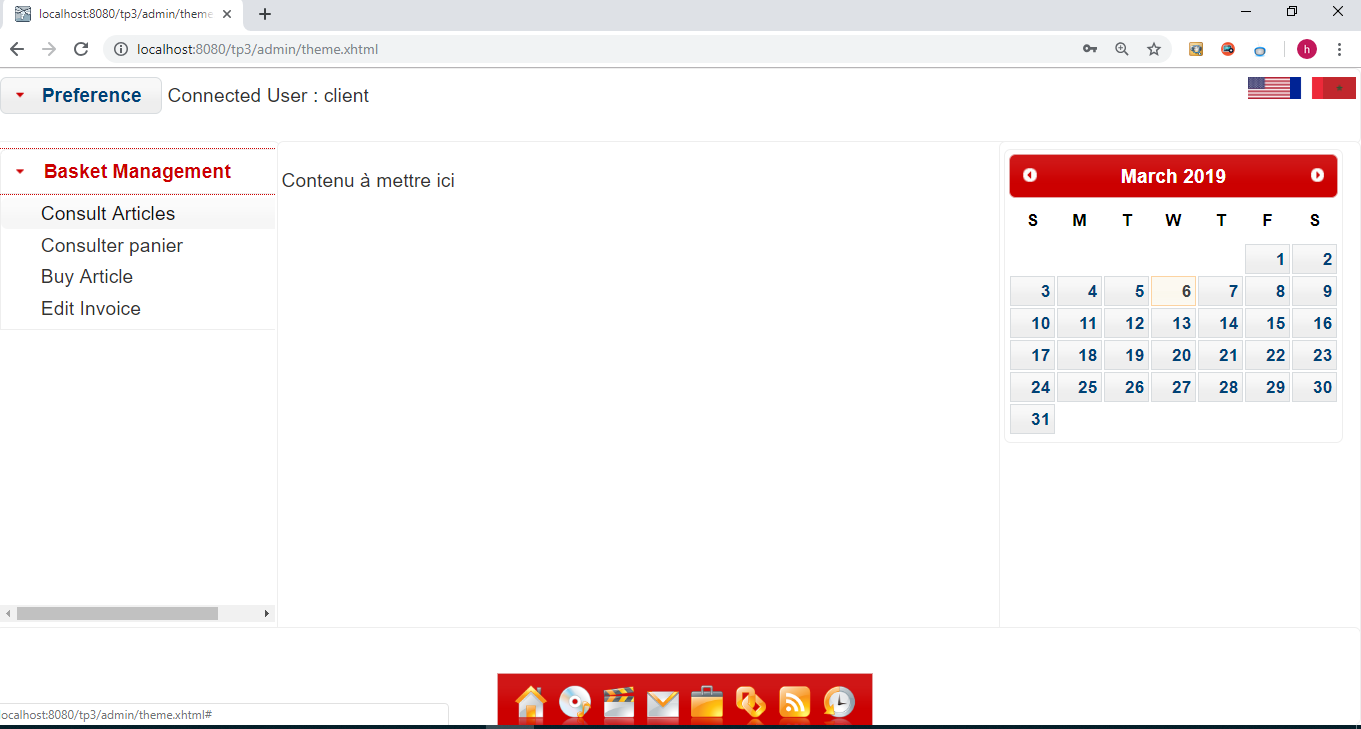
**\*** Pour se déconnecter, cliquer sur le lien suivant :



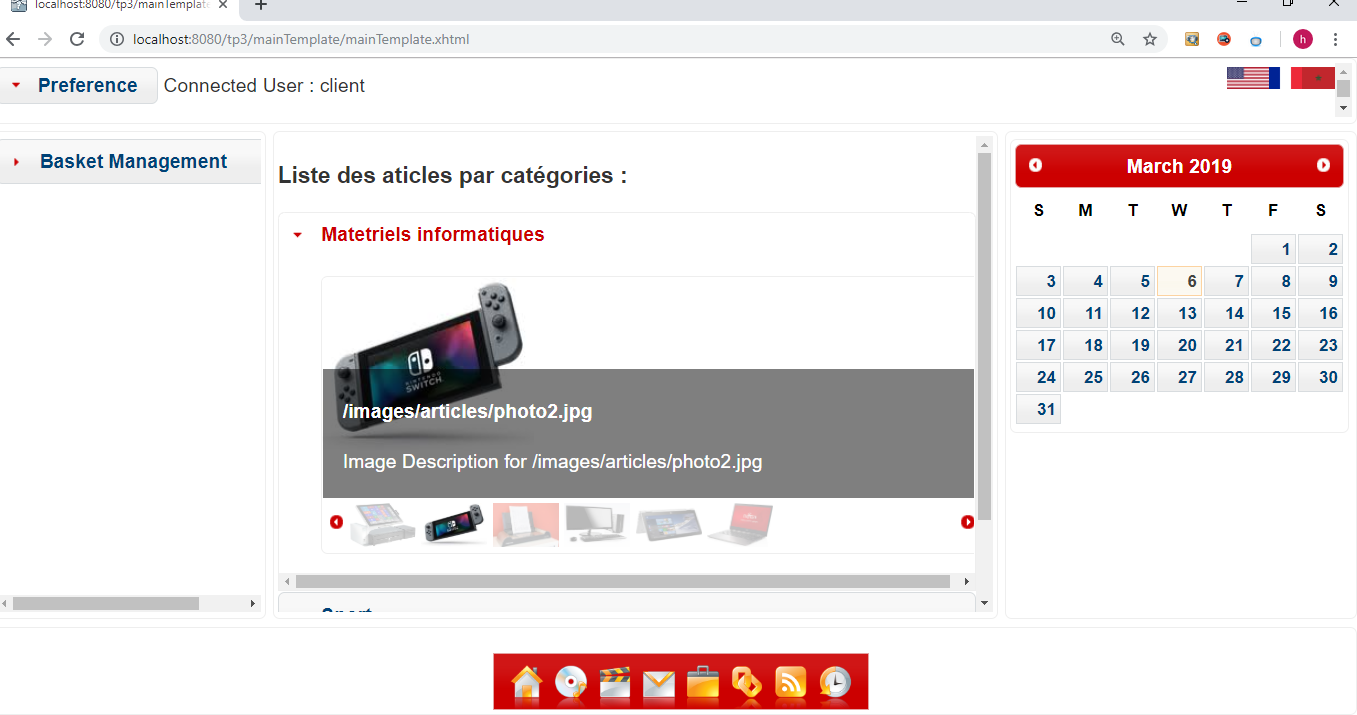
La page suivante sera affichée :



\* Connectez vous avec un compte client (clien/client par exemple). La page suivante sera affichée :



\* Pour consulter les articles, cliquer sur "Consult Articles" :



**Exercices :**

Reprendre le TP4 et implémenter les fonctionnalités suivantes :

* En tant que client (profile=CLIENT), chercher les articles par catégorie, par description et par prix.
* Ajouter les articles au panier en spécifiant la quantité à acheter de chaque article. Il faut s'assurer que la quantité demandée existe dans le stock.
* Consulter et éditer le panier.
* Editer la facture en format PDF.