|  |
| --- |
| Formation JSF par la pratique  Partie 2 : Gestion des sessions avec JSF |
| Développement d'une application web avec JSF, JPA et WildFly |

Sommaire

[1. Objet 3](#_Toc1755166)

[2. Pré requis 3](#_Toc1755167)

[3. Partie 1 : Préparation de l'environnement de développement 3](#_Toc1755168)

[4. Partie 2 : Développement de l'application web 3](#_Toc1755169)

[**A.** **Le modèle :** 7](#_Toc1755170)

[**B.** **La couche DAO :** 11](#_Toc1755171)

[**C.** **La couche domaine :** 19](#_Toc1755172)

[**D.** **La couche service :** 25](#_Toc1755173)

[**E.** **La couche présentation** 30](#_Toc1755174)

[**F.** **Les fichiers messages\_x.properties** 36](#_Toc1755175)

[**G.** **Le fichier persistence.xml :** 39](#_Toc1755176)

[**H.** **Les pages \*.xhtml** 40](#_Toc1755177)

[**I.** **La feuille styles.css** 45](#_Toc1755178)

[**J.** **Les images** 46](#_Toc1755179)

[**K.** **Le fichier faces-config.xml** 46](#_Toc1755180)

[**L.** **Le fchier web.xml** 46](#_Toc1755181)

[**M.** **Les classes de test** 46](#_Toc1755182)

[5. Les tests : 48](#_Toc1755183)

# Objet

Le présent document a pour objet de vous montrer comment gérer les sessions avec l'api JSF (Java Server Faces), comment valider les champs du formulaire, comment gérer les messageset comment gérer les évenements.

# Pré requis

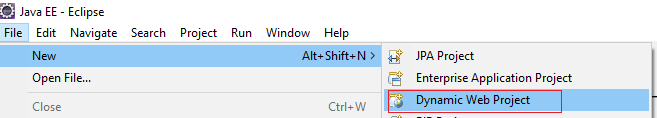
Le TP n°1 (partie 1) doit être réalisé avant. Les mêmes outils que le TP n°1.

# Partie 1 : Préparation de l'environnement de développement

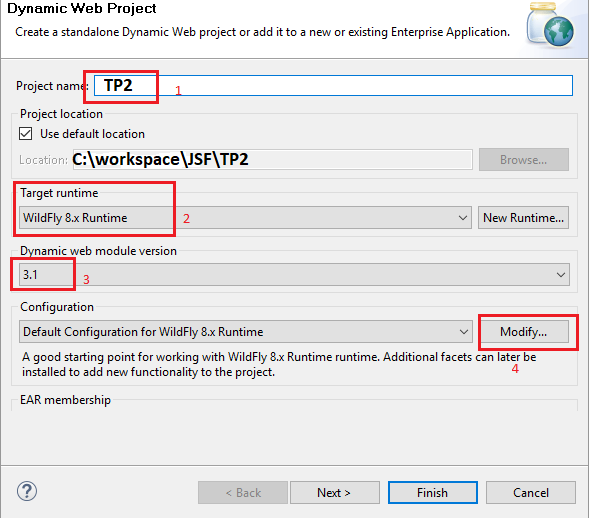
Idem que le TP n°1.

# Partie 2 : Développement de l'application web

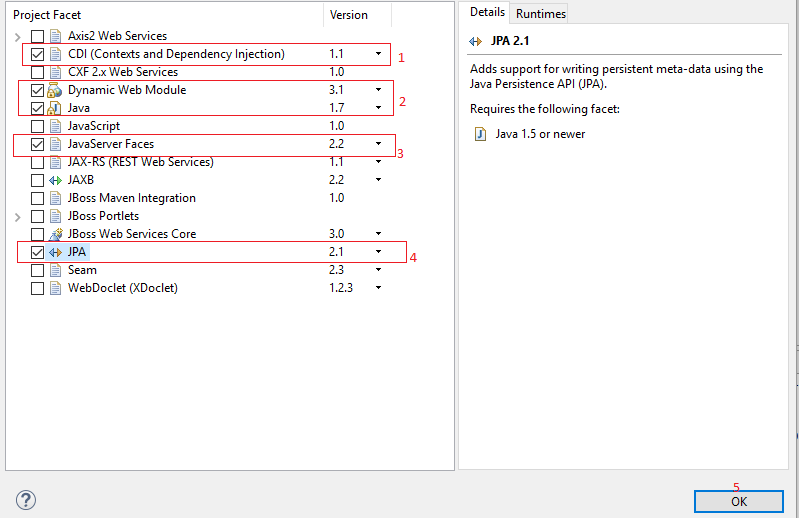
Au niveau d'Eclipse, créer un projet web comme illustre la fenêtre suivante :



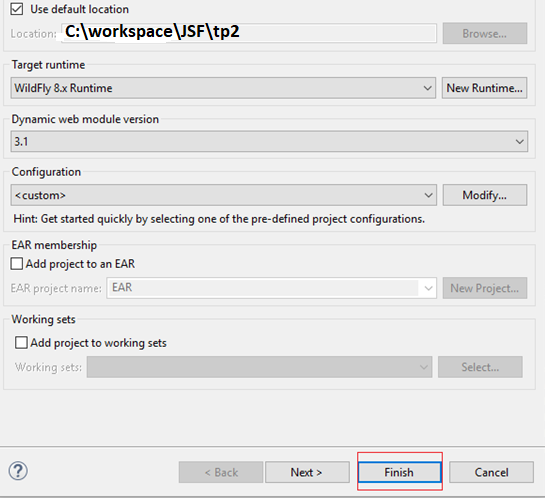
Cliquer sur "Dynamic Web Project" :



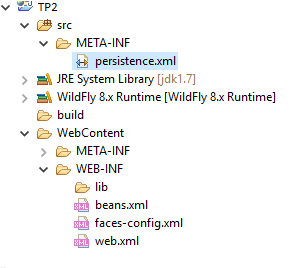
1. Saisir le nom de votre projet (ici : TP2);
2. Préciser la "Target runtime" Wildfly 8.x Runtime;
3. Préciser la version de la Servlet (ici : 3.1) ;
4. Cliquer ensuite sur Modifiy.... L'écran suivant sera affiché :



1. Cocher CDI (Context and Dependency Injection). L'api CDI implémente le Design Pattern IOC, nous en parlerons par la suite.
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 :



1. **Le modèle :**

La classe EntityMere :

|  |
| --- |
| **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** EntityMere **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.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.Table;  @Entity  @Table(name = "Article")  **public** **class** Article **extends** EntityMere **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 6161046340508532663L;  @Override  **public** String toString() {  **return** "Article [id=" + id + ", designation=" + designation + ", prix="  + prix + ", quantiteEnStock=" + quantiteEnStock + "]";  }  @Column(name = "libelle")  **private** String designation;  **private** Double prix;  **private** Integer quantiteEnStock;  **public** Article() {  }  **public** Article(String designation, Double prix, Integer quantiteEnStock) {  **super**();  **this**.designation = designation;  **this**.prix = prix;  **this**.quantiteEnStock = quantiteEnStock;  }  **public** Article(Integer id, String designation, Double prix,  Integer quantiteEnStock) {  **this**(designation, prix, quantiteEnStock);  **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 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** EntityMere **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() {  }  **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** EntityMere **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(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. **La couche DAO :**

**L'interface IBaseDao :**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao;  **import** javax.persistence.EntityManager;  **public** **interface** IBaseDao {    **void** beginTransaction();  **void** validTransaction();  **void** closeSession();  <T> **void** save(T t);  **void** setSession(EntityManager session);  EntityManager getSession();  } |

**L'interface IUserDao :**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.User;  **public** **interface** IUserDao **extends** IBaseDao {  User getByLogin(String login);  **void** remove(Integer id);  List<User> getAll();  User getById(Integer id);  } |

**L'interface IArticleDao :**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.Article;  **public** **interface** IArticleDao **extends** IBaseDao{  **void** remove(Integer id);  List<Article> getAll();  **public** Article getById(Integer id);  } |

**La classe BaseDaoImpl (Implémentation List):**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao;  **import** javax.persistence.EntityManager;  **public** **class** BaseDaoImpl **implements** IBaseDao {  @Override  **public** **void** beginTransaction() {  }  @Override  **public** **void** validTransaction() {  }  @Override  **public** **void** closeSession() {  }  @Override  **public** <T> **void** save(T t) {  }  @Override  **public** **void** setSession(EntityManager session) {  }  @Override  **public** EntityManager getSession() {  **return** **null**;  }  } |

**La classe UserDaoImpl (Imlémentation List):**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.presentation.utils.Constants;  **import** ma.gov.formation.cigma.service.modele.User;  **import** ma.gov.formation.cigma.service.modele.UserRole;  /\*\*  \* Cette classe permet de stocker les articles dans une liste static.  \*/  **public** **class** UserDaoImpl **extends** BaseDaoImpl **implements** IUserDao {  **private** **static** List<User> *modeles* = **new** ArrayList<User>();  **private** **static** **int** *count* = 0;  **static** {  UserRole adminRole = **new** UserRole(1, Constants.***ADMIN\_ROLE***,  Constants.***ADMIN\_ROLE\_DESCRIPTION***);  UserRole clientRole = **new** UserRole(2, Constants.***CLIENT\_ROLE***,  Constants.***CLIENT\_ROLE\_DESCRIPTION***);  List<UserRole> admin = **new** ArrayList<>();  admin.add(adminRole);  List<UserRole> client = **new** ArrayList<>();  client.add(clientRole);  List<UserRole> all = **new** ArrayList<>();  all.add(adminRole);  all.add(clientRole);  *modeles*.add(**new** User(1, "admin", "admin", admin));  *modeles*.add(**new** User(2, "client", "client", client));  *modeles*.add(**new** User(3, "superadmin", "superadmin", all));  *count* = 3;  }  @Override  **public** <T> **void** save(T t) {  User user=(User)t;  **if** (exists(user.getId())) {  User a = getById(user.getId());  *modeles*.remove(a);  *modeles*.add(user);  **return**;  }  *count*++;  user.setId(*count*);  *modeles*.add(user);  }  @Override  **public** **void** remove(Integer id) {  **if** (exists(id)) {  User a = getById(id);  *modeles*.remove(a);  *count*--;  }  }  @Override  **public** List<User> getAll() {  **return** *modeles*;  }  @Override  **public** User getById(Integer id) {  **if** (id == **null**)  **return** **null**;  **for** (User modele : *modeles*) {  **if** (id.equals(modele.getId())) {  **return** modele;  }  }  **return** **null**;  }  **private** **boolean** exists(Integer id) {  **if** (id == **null**)  **return** **false**;  **for** (User modele : *modeles*)  **if** (id.equals(modele.getId())) {  **return** **true**;  }  **return** **false**;  }  @Override  **public** User getByLogin(String login) {  **if** (login == **null** || login.equalsIgnoreCase(""))  **return** **null**;  **for** (User user : *modeles*) {  **if** (login.equals(user.getUserName())) {  **return** user;  }  }  **return** **null**;  }  } |

**La classe AticleDaoImpl (Implémentation List):**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.Article;  /\*\*  \* Cette classe permet de stocker les articles dans une liste static.  \*/  **public** **class** ArticleDaoImpl **extends** BaseDaoImpl **implements** IArticleDao {  **private** **static** List<Article> *modeles* = **new** ArrayList<Article>();  **private** **static** **int** *count* = 0;  **static** {  *modeles*.add(**new** Article(1, "PC SONY", 15000.0, 20));  *modeles*.add(**new** Article(2, "TV LG 32 p", 4000.0, 20));  *modeles*.add(**new** Article(3, "TV Samsung 43 p", 65000.0, 35));  *modeles*.add(**new** Article(4, "Camera Sony", 5000.0, 18));  *count* = 4;  }  @Override  **public** <T> **void** save(T t) {  Article article = (Article) t;  **if** (exists(article.getId())) {  Article a = getById(article.getId());  *modeles*.remove(a);  *modeles*.add(article);  **return**;  }  *count*++;  article.setId(*count*);  *modeles*.add(article);  }  @Override  **public** **void** remove(Integer id) {  **if** (exists(id)) {  Article a = getById(id);  *modeles*.remove(a);  *count*--;  }  }  @Override  **public** List<Article> getAll() {  **return** *modeles*;  }  @Override  **public** Article getById(Integer id) {  **if** (id == **null**)  **return** **null**;  **for** (Article modele : *modeles*) {  **if** (id.equals(modele.getId())) {  **return** modele;  }  }  **return** **null**;  }  **private** **boolean** exists(Integer id) {  **if** (id == **null**)  **return** **false**;  **for** (Article modele : *modeles*)  **if** (id.equals(modele.getId())) {  **return** **true**;  }  **return** **false**;  }  } |

**La classe SessionBuilder :**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao.jpa;  **import** javax.persistence.EntityManager;  **import** javax.persistence.EntityManagerFactory;  **import** javax.persistence.Persistence;  /\*\*  \* 1- Cette classe permet de créer la factory de JPA (EntityManagerFactory) et  \* par la suite ouvrir une session JPA.  \*  \* 2- Dans cette classe, nous avons implémenté le Singleton afin d'avoir une  \* session instance de la factory.  \*/  **public** **class** SessionBuilder {  **private** **static** EntityManagerFactory *sessionFactory* = **null**;  **private** **static** SessionBuilder *instance* = **null**;  **private** SessionBuilder() {  }  **public** **static** SessionBuilder getInstance() {  **if** (*instance* == **null**) {  **try** {  *instance* = **new** SessionBuilder();  *sessionFactory* = Persistence  .*createEntityManagerFactory*("unite1");  } **catch** (Exception e) {  System.***err***.println("session factory error " + e);  e.printStackTrace();  }  }  **return** *instance*;  }  **public** EntityManager getSession() {  **return** *sessionFactory*.createEntityManager();  }  } |

**La classe BaseDaoImpl (L'implémentation JPA):**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao.jpa;  **import** javax.persistence.EntityManager;  **import** javax.persistence.EntityTransaction;  **import** ma.gov.formation.cigma.dao.IBaseDao;  /\*\*  \* Cette classe utilise JPA pour ouvrir et fermer la session et également pour  \* débuter et valider la transaction..  \*/  **public** **class** BaseDaoImpl **implements** IBaseDao {  **protected** EntityManager session = **null**;  **protected** EntityTransaction tx = **null**;  @Override  **public** **void** beginTransaction() {  session = SessionBuilder.*getInstance*().getSession();  tx = session.getTransaction();  tx.begin();  }  @Override  **public** **void** validTransaction() {  tx.commit();  }  @Override  **public** **void** closeSession() {  session.close();  }  @Override  **public** <T> **void** save(T t) {  session.merge(t);  }  @Override  **public** **void** setSession(EntityManager session) {  **this**.session = session;  }  @Override  **public** EntityManager getSession() {  **return** session;  }  } |

**La classe ArticleDaoImpl (la deuxième implémentation avec JPA) :**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao.jpa;  **import** java.util.List;  **import** ma.gov.formation.cigma.dao.IArticleDao;  **import** ma.gov.formation.cigma.service.modele.Article;  /\*\*  \* Cette classe utiliser JPA pour stocker les articles.  \*/  **public** **class** ArticleDaoImpl **extends** BaseDaoImpl **implements** IArticleDao {  @SuppressWarnings("unchecked")  **public** List<Article> getAll() {  List<Article> liste = **null**;  liste = (List<Article>) session.createQuery("from Article")  .getResultList();  **return** liste;  }  @Override  **public** **void** remove(Integer id) {  Article a = session.find(Article.**class**, id);  **if** (a != **null**)  session.remove(a);  }  @Override  **public** Article getById(Integer id) {  **return** session.find(Article.**class**, id);  }  } |

**La classe UserDaoImpl (Implémentation JPA):**

|  |
| --- |
| **package** ma.gov.formation.cigma.dao.jpa;  **import** java.util.List;  **import** javax.persistence.Query;  **import** ma.gov.formation.cigma.dao.IUserDao;  **import** ma.gov.formation.cigma.service.modele.User;  /\*\*  \* Cette classe utiliser JPA pour stocker les utilisateurs.  \*/  **public** **class** UserDaoImpl **extends** BaseDaoImpl **implements** IUserDao {  @Override  @SuppressWarnings("unchecked")  **public** User getByLogin(String userName) {  User a = **null**;  Query q = session.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;  }  @SuppressWarnings("unchecked")  **public** List<User> getAll() {  List<User> liste = **null**;  liste = (List<User>) session.createQuery("from User")  .getResultList();  **return** liste;  }  @Override  **public** **void** remove(Integer id) {  User a = session.find(User.**class**, id);  **if** (a != **null**)  session.remove(a);  }  @Override  **public** User getById(Integer id) {  **return** session.find(User.**class**, id);  }  } |

1. **La couche domaine :**

**La classe ArticleVo :**

|  |
| --- |
| **package** ma.gov.formation.cigma.domaine.article;  **import** java.io.Serializable;  /\*\*  \* Cette classe est le Value Object (VO). C'est cette classe que nous allons  \* utiliser dans la couche présentation. Le VO nous permet de pas utiliser les  \* objects externes dans la couche présentation.  \*  \*/  **public** **class** ArticleVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 6161046340508532663L;  **private** Integer id;  **private** String designation;  **private** Double prix;  **private** Integer quantiteEnStock;  **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;  }  **public** Integer getId() {  **return** id;  }  **public** **void** setId(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;  }  **public** **void** init() {  id = **null**;  designation = **null**;  prix = **null**;  quantiteEnStock = **null**;  editable=**false**;  }  } |

**La classe ArticleConverter :**

|  |
| --- |
| **package** ma.gov.formation.cigma.domaine.article;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.Article;  /\*\*  \* Cette classe permet de gérer le pont VO <-> BO. Donc, si il y a une  \* modification au niveau du BO en change unqiuement dans cette classe sans  \* impacter le VO. De cette façon, le couplage sera faible.  \*/  **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());  **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());  **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 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.user.role.UserRoleVo;  /\*\*  \* Cette classe est le Value Object (VO). C'est cette classe que nous allons  \* utiliser dans la couche présentation. Le VO nous permet de pas utiliser les  \* objects externes dans la couche présentation.  \*  \*/  **public** **class** UserVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = -5500475381869205248L;  **private** Integer id;  **private** String userName;  **private** String password;  **private** List<UserRoleVo> roles = **new** ArrayList<>();  **public** List<UserRoleVo> getRoles() {  **return** roles;  }  **public** **void** setRoles(List<UserRoleVo> roles) {  **this**.roles = roles;  }  **public** Integer getId() {  **return** id;  }  **public** UserVo() {  **super**();  // **TODO** Auto-generated constructor stub  }  /\*\*  \* **@param** userName  \* **@param** password  \*/  **public** UserVo(String userName, String password) {  **super**();  **this**.userName = userName;  **this**.password = password;  }  /\*\*  \* **@param** id  \* **@param** userName  \* **@param** password  \*/  **public** UserVo(Integer id, String userName, String password) {  **super**();  **this**.id = id;  **this**.userName = userName;  **this**.password = password;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **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.getRole()))  **return** **true**;  }  **return** result;  }  } |

**La classe UserRoleVo :**

|  |
| --- |
| **package** ma.gov.formation.cigma.domaine.user.role;  **import** java.io.Serializable;  /\*\*  \* Cette classe est le Value Object (VO). C'est cette classe que nous allons  \* utiliser dans la couche présentation. Le VO nous permet de pas utiliser les  \* objects externes dans la couche présentation.  \*  \*/  **public** **class** UserRoleVo **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 4928414063636905875L;  **private** Integer id;  **private** String role;  **private** String libelle;  **public** UserRoleVo() {  **super**();  // **TODO** Auto-generated constructor stub  }    /\*\*  \* **@param** role  \* **@param** libelle  \*/  **public** UserRoleVo(String role, String libelle) {  **super**();  **this**.role = role;  **this**.libelle = libelle;  }  /\*\*  \* **@param** id  \* **@param** role  \* **@param** libelle  \*/  **public** UserRoleVo(Integer id, String role, String libelle) {  **super**();  **this**.id = id;  **this**.role = role;  **this**.libelle = libelle;  }  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **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;  }  } |

**La classe UserRoleConverter :**

|  |
| --- |
| **package** ma.gov.formation.cigma.domaine.user.role;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.service.modele.UserRole;  /\*\*  \* Cette classe permet de gérer le pont VO <-> BO. Donc, si il y a une  \* modification au niveau du BO en change unqiuement dans cette classe sans  \* impacter le VO. De cette façon, le couplage sera faible.  \*/  **public** **class** UserRoleConverter {  **public** **static** UserRole createModele(UserRoleVo vo) {  UserRole modele = **new** UserRole();  modele.setId(vo.getId());  modele.setLibelle(vo.getLibelle());  modele.setRole(vo.getRole());  **return** modele;  }  **public** **static** UserRoleVo createUserRoleVo(UserRole modele) {  UserRoleVo vo = **new** UserRoleVo();  vo.setId(modele.getId());  vo.setLibelle(modele.getLibelle());  vo.setRole(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;  }  } |

**La Classe UserConverter :**

|  |
| --- |
| **package** ma.gov.formation.cigma.domaine.user;  **import** java.util.ArrayList;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.user.role.UserRoleConverter;  **import** ma.gov.formation.cigma.domaine.user.role.UserRoleVo;  **import** ma.gov.formation.cigma.service.modele.User;  /\*\*  \* Cette classe permet de gérer le pont VO <-> BO. Donc, si il y a une  \* modification au niveau du BO en change unqiuement dans cette classe sans  \* impacter le VO. De cette façon, le couplage sera faible.  \*/  **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> rolesVo=vo.getRoles();  **if** (rolesVo == **null** || rolesVo.isEmpty())  **return** modele;  **for** (UserRoleVo userRoleVo : rolesVo) {  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;  }  } |

1. **La couche service :**

**L'interface IBaseService :**

|  |
| --- |
| **package** ma.gov.formation.cigma.service;  **public** **interface** IBaseService {  **void** beginTransaction();  **void** validTransaction();  **void** closeSession();  } |

**L'interface IUserService :**

|  |
| --- |
| **package** ma.gov.formation.cigma.service;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  **public** **interface** IUserService **extends** IBaseService{  **public** **void** save(UserVo a);  **public** UserVo getById(Integer id);  **public** List<UserVo> getAll();  **public** **void** remove(Integer id);  **public** UserVo getByLogin(String login);  } |

**L'interface IArticleService :**

|  |
| --- |
| **package** ma.gov.formation.cigma.service;  **import** java.util.List;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  **public** **interface** IArticleService **extends** IBaseService {  **public** **void** save(ArticleVo a);  **public** ArticleVo getById(Integer id);  **public** List<ArticleVo> getAll();  **public** **void** remove(Integer id);  } |

**La classe BaseServiceImpl :**

|  |
| --- |
| **package** ma.gov.formation.cigma.service;  **import** ma.gov.formation.cigma.dao.IBaseDao;  **import** ma.gov.formation.cigma.dao.jpa.BaseDaoImpl;  **public** **class** BaseServiceImpl **implements** IBaseService {  //L'implémentation JPA :ma.gov.formation.cigma.dao.jpa.BaseDaoImpl  **protected** IBaseDao baseDao=**new** BaseDaoImpl();    //L'implémentation List :ma.gov.formation.cigma.dao.BaseDaoImpl  // protected IBaseDao baseDao=new BaseDaoImpl();  @Override  **public** **void** beginTransaction() {  baseDao.beginTransaction();  }  @Override  **public** **void** validTransaction() {  baseDao.validTransaction();  }  @Override  **public** **void** closeSession() {  baseDao.closeSession();    }  } |

**La classe ArticleServiceImpl :**

|  |
| --- |
| **package** ma.gov.formation.cigma.service;  **import** java.util.List;  **import** ma.gov.formation.cigma.dao.IArticleDao;  **import** ma.gov.formation.cigma.dao.jpa.ArticleDaoImpl;  **import** ma.gov.formation.cigma.domaine.article.ArticleConverter;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  /\*\*  \* 1- Tous les services de la classe ArticleServiceImpl sont transactionnelles.  \* Dans les bonnes pratiques, les transactions doivent être gérées dans la  \* couche métier.  \*  \* 2- Dans cette couche, nous avons implémenté le Value Object (VO). Cette bonne  \* pratique nous permet de ne pas changer la couche présentaion si on change les  \* modèles de la couche métier. Le VO nous permet également de régler le  \* problème d'accéder à un objet qui n'est pas attaché à la session JPA.  \*  \* 3- La mauvaise pratique au niveau de cette classe est le fait d'instancier  \* l'objet de type ArticleDaoImpl. Normlament ce qui est recommandé par SUN est  \* que le couplage doit être faible entre les classes. Nous allons résoudre ce  \* problème par la suite lorsque nous attaquerons le framework Spring.Spring  \* implémente l'IOC moyennant son module Spring IOC.  \*  \* 4- La deuxième mauvaise pratique au niveau de cette classe est le fait que  \* nous avons mélangé deux types de logiques :la logique métier et la logique  \* technique qui est celle de la gestion des transactions. L'AOP est la bonne  \* pratqiue qui résouds ce problème. Nous allons l'aborder par la suite lorsque  \* nous allons aborder Spring AOP.  \*/  **public** **class** ArticleServiceImpl **extends** BaseServiceImpl **implements**  IArticleService {  // L'implémentation JPA :  **private** IArticleDao dao = **new** ArticleDaoImpl();  @Override  **public** **void** save(ArticleVo vo) {  **try** {  beginTransaction();  dao.save(ArticleConverter.*createModele*(vo));  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  }  }  @SuppressWarnings("finally")  @Override  **public** ArticleVo getById(Integer id) {  ArticleVo vo = **null**;  **try** {  beginTransaction();  vo = ArticleConverter.*createArticleVo*(dao.getById(id));  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  **return** vo;  }  }  @SuppressWarnings("finally")  @Override  **public** List<ArticleVo> getAll() {  List<ArticleVo> list = **null**;  **try** {  beginTransaction();  list = ArticleConverter.*getArticleVoList*(dao.getAll());  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  **return** list;  }  }  @Override  **public** **void** remove(Integer id) {  **try** {  beginTransaction();  dao.remove(id);  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  }  }  @Override  **public** **void** beginTransaction() {  **super**.beginTransaction();  dao.setSession(baseDao.getSession());  }  } |

**La classe UserServiceImpl :**

|  |
| --- |
| **package** ma.gov.formation.cigma.service;  **import** java.util.List;  **import** ma.gov.formation.cigma.dao.IUserDao;  **import** ma.gov.formation.cigma.dao.jpa.UserDaoImpl;  **import** ma.gov.formation.cigma.domaine.user.UserConverter;  **import** ma.gov.formation.cigma.domaine.user.UserVo;  /\*\*  \* 1- Tous les services de la classe UserServiceImpl sont transactionnelles.  \* Dans les bonnes pratiques, les transactions doivent être gérées dans la  \* couche métier.  \*  \* 2- Dans cette couche, nous avons implémenté le Value Object (VO). Cette bonne  \* pratique nous permet de ne pas changer la couche présentaion si on change les  \* modèles de la couche métier. Le VO nous permet également de régler le  \* problème d'accéder à un objet qui n'est pas attaché à la session JPA.  \*  \* 3- La mauvaise pratique au niveau de cette classe est le fait d'instancier  \* l'objet de type ArticleDaoImpl. Normlament ce qui est recommandé par SUN est  \* que le couplage doit être faible entre les classes. Nous allons résoudre ce  \* problème par la suite lorsque nous attaquerons le framework Spring.Spring  \* implémente l'IOC moyennant son module Spring IOC.  \*  \* 4- La deuxième mauvaise pratique au niveau de cette classe est le fait que  \* nous avons mélangé deux types de logiques :la logique métier et la logique  \* technique qui est celle de la gestion des transactions. L'AOP est la bonne  \* pratqiue qui résouds ce problème. Nous allons l'aborder par la suite lorsque  \* nous allons aborder Spring AOP.  \*/  **public** **class** UserServiceImpl **extends** BaseServiceImpl **implements** IUserService {  // L'implémentation JPA :  **private** IUserDao dao = **new** UserDaoImpl();  @Override  **public** **void** save(UserVo vo) {  **try** {  beginTransaction();  dao.save(UserConverter.*createModele*(vo));  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  }  }  @SuppressWarnings("finally")  @Override  **public** UserVo getById(Integer id) {  UserVo vo = **null**;  **try** {  beginTransaction();  vo = UserConverter.*createUserVo*(dao.getById(id));  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.validTransaction();  **return** vo;  }  }  @SuppressWarnings("finally")  @Override  **public** List<UserVo> getAll() {  List<UserVo> list = **null**;  **try** {  beginTransaction();  list = UserConverter.*getUserVoList*(dao.getAll());  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  **return** list;  }  }  @Override  **public** **void** remove(Integer id) {  **try** {  beginTransaction();  dao.remove(id);  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  }  }  @SuppressWarnings("finally")  @Override  **public** UserVo getByLogin(String login) {  UserVo vo = **null**;  **try** {  beginTransaction();  vo = UserConverter.*createUserVo*(dao.getByLogin(login));  validTransaction();  } **catch** (Exception e) {  e.printStackTrace();  } **finally** {  dao.closeSession();  **return** vo;  }  }    @Override  **public** **void** beginTransaction() {  **super**.beginTransaction();  dao.setSession(baseDao.getSession());  }  } |

1. **La couche présentation**

**La classe BaseAction :**

|  |
| --- |
| **package** ma.gov.formation.cigma.presentation;  **import** java.util.ResourceBundle;  **import** javax.faces.application.FacesMessage;  **import** javax.faces.bean.ManagedProperty;  **import** javax.faces.context.FacesContext;  **import** ma.gov.formation.cigma.presentation.utils.Constants;  **public** **class** BaseAction {  @ManagedProperty("#{msg}")  **protected** ResourceBundle bundle;  **public** ResourceBundle getBundle() {  **return** bundle;  }  **public** **void** setBundle(ResourceBundle bundle) {  **this**.bundle = bundle;  }  **public** **void** addMessage(String message, String severity) {  FacesMessage msg = **null**;  msg = **new** FacesMessage(bundle.getString(message));  **if** (severity.equals(Constants.***ERROR\_MSG***))  msg.setSeverity(FacesMessage.***SEVERITY\_ERROR***);  **if** (severity.equals(Constants.***INFO\_MSG***))  msg.setSeverity(FacesMessage.***SEVERITY\_INFO***);    FacesContext.*getCurrentInstance*().addMessage(**null**, msg);  }  } |

**La classe Utils :**

|  |
| --- |
| **package** ma.gov.formation.cigma.presentation.utils;  **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.utils;  /\*\*  \* Nous gérons deux type de profiles :  \* 1- Un profile administrateur dont le code est "ADMIN\_ROLE".  \* 2- Un profile client dont le code est "CLIENT\_ROLE"  \*/  **public** **interface** Constants {  //Les codes des roles :  **public** **static** **final** String ***ADMIN\_ROLE***="ADMIN";  **public** **static** **final** String ***CLIENT\_ROLE***="CLIENT";    //La description des roles :  **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";  } |

**La classe ArticleBean :**

|  |
| --- |
| **package** ma.gov.formation.cigma.presentation.article;  **import** java.util.ArrayList;  **import** java.util.List;  **import** javax.faces.bean.ManagedBean;  **import** javax.faces.bean.SessionScoped;  **import** ma.gov.formation.cigma.domaine.article.ArticleVo;  **import** ma.gov.formation.cigma.presentation.BaseAction;  **import** ma.gov.formation.cigma.presentation.utils.Constants;  **import** ma.gov.formation.cigma.service.ArticleServiceImpl;  **import** ma.gov.formation.cigma.service.IArticleService;  /\*\*  \* 1- Il s'agit du Managed Bean qui permet de gérer les CRUD des articles. Une  \* instance de ce bean sera mis dans le scope "Session". De cette façon,  \* l'utilisateur pourra accéder aux articles pendant sa session.  \*  \* 2- Dans cette classe, nous avons utilisé le VO et non pas les modèles de la  \* couche métier et ceci pour réduire le couplage entre les couches.  \*  \* 3- La relation entre la couche présentation et la couche métier est  \* IArticleService. Il s'agit là d'une bonne pratique.  \*  \* 4- La mauvaise pratique est que nous avons instancié un objet de type  \* ArticleServiceImpl. Nous allons régler ce promlème par la suite lorsque nous  \* allons aborder IOC.  \*/  @ManagedBean(name = "articleBean")  @SessionScoped  **public** **class** ArticleBean **extends** BaseAction{  **private** IArticleService service = **new** ArticleServiceImpl();  **private** List<ArticleVo> articlesVo = **new** ArrayList<>();  **private** ArticleVo articleVo = **new** ArticleVo();  **private** **boolean** editable = **false**;  **public** **boolean** isEditable() {  **return** editable;  }  **public** **void** setEditable(**boolean** editable) {  **this**.editable = editable;  }  **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() {  **if** (!editable)  articlesVo = service.getAll();  **return** articlesVo;  }  **public** String addArticle() {  editable=**false**;  service.save(articleVo);  articleVo.init();  addMessage("article.ajoute.avec.succes", Constants.***INFO\_MSG***);  **return** **null**;  }  **public** String editArticle(Integer id) {  editable = **true**;  **for** (ArticleVo vo : articlesVo) {  **if** (id.equals(vo.getId()))  vo.setEditable(**true**);  }  **return** **null**;  }  **public** String removeArticle(Integer id) {  editable=**false**;  service.remove(id);  articleVo.init();  addMessage("article.sypprime.avec.succes", Constants.***INFO\_MSG***);  **return** **null**;  }  **public** String saveUpdate() {  editable = **false**;  service.save(articlesVo);  addMessage("article.modifier.avec.succes", Constants.***INFO\_MSG***);  **return** **null**;  }    **public** String cancelUpdate() {  editable = **false**;  **return** **null**;  }  } |

**La classe LoginBean :**

|  |
| --- |
| **package** ma.gov.formation.cigma.presentation.authentication;  **import** javax.faces.application.ConfigurableNavigationHandler;  **import** javax.faces.bean.ManagedBean;  **import** javax.faces.bean.SessionScoped;  **import** javax.faces.context.FacesContext;  **import** javax.faces.event.ComponentSystemEvent;  **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.utils.Constants;  **import** ma.gov.formation.cigma.presentation.utils.Utils;  **import** ma.gov.formation.cigma.service.IUserService;  **import** ma.gov.formation.cigma.service.UserServiceImpl;  /\*\*  \* 1- Il s'agit du Managed Bean qui permet de gérer l'authentification des  \* utilisateurs.  \*  \* 2- Dans cette classe, nous avons utilisé le VO et non pas les modèles de la  \* couche métier et ceci pour réduire le couplage entre les couches.  \*  \* 3- La relation entre la couche présentation et la couche métier est  \* IUserService. Il s'agit là d'une bonne pratique.  \*  \* 4- La mauvaise pratique est que nous avons instancié un objet de type  \* UserServiceImpl. Nous allons régler ce problème par la suite lorsque nous  \* allons aborder l'IOC.  \*  \*/  @ManagedBean(name = "loginBean")  @SessionScoped  **public** **class** LoginBean **extends** BaseAction {  **private** IUserService service = **new** UserServiceImpl();  **private** UserVo userVo = **new** UserVo();  **public** UserVo getUserVo() {  **return** userVo;  }  // logout event, invalidate session  **public** String logout() {  HttpSession session = Utils.*getSession*();  session.invalidate();  **return** "login";  }  **public** **void** setUserVo(UserVo userVo) {  **this**.userVo = userVo;  }  /\*\*  \* 1- Si l'utilisateur ne s'est pas authentifié, l'application va le  \* rediriger vers la page login.xhtml  \*  \* 2- Si l'ulisateur qui est connecté dispose du rôle "ADMIN\_ROLE",  \* l'application va lui rediriger vers la page articles.xhtml et si il  \* dispose du rôle "CLIENT\_ROLE", l'application va lui rediriger vers la  \* page panier.xhtml.  \*/  **public** String login() {  UserVo userVoP = service.getByLogin(userVo.getUserName());  **if** (userVoP == **null**) {  addMessage("login.doesnt.exist", Constants.***ERROR\_MSG***);  **return** "login";  }  **if** (!userVo.getPassword().equals(userVoP.getPassword())) {  addMessage("password.incorrect", Constants.***ERROR\_MSG***);  **return** "login";  }  userVo=userVoP;  HttpSession session = Utils.*getSession*();  session.setAttribute("user", userVo);  **if** (userVoP.haseRole(Constants.***ADMIN\_ROLE***)) {  **return** "articles";  }  **if** (userVoP.haseRole(Constants.***CLIENT\_ROLE***)) {  **return** "panier";  }  addMessage("problem.user.role", Constants.***ERROR\_MSG***);  **return** "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("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("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.internationalisation;  **import** java.io.Serializable;  **import** java.util.Locale;  **import** javax.faces.bean.ManagedBean;  **import** javax.faces.bean.SessionScoped;  **import** javax.faces.context.FacesContext;  **import** ma.gov.formation.cigma.presentation.BaseAction;  /\*\*  \* 1- Il s'agit du Managed Bean qui permet de gérer l'internationnalisation. Une  \* instance de ce bean sera mis dans le scope "Session". De cette façon,  \* l'utilisateur pourra accéder à sa langue qui est choisie par lui même.  \*  \* 2- Pour changer la langue au niveau de JSF, il suffit d'uliser la classe  \* FacesContext comme suit:  \* FacesContext.getCurrentInstance().getViewRoot().setLocale(locale).  \*/  @ManagedBean(name = "langue")  @SessionScoped  **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);  }  } |

1. **Les fichiers messages\_x.properties**

**Le fichier messages\_en.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  must.authentify=You shoud authentify with your account.  #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  #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=Save update  cancel.update=Cancel update  #Infos messages  article.ajoute.avec.succes=Article added with success.  article.sypprime.avec.succes=Article removed with success.  article.modifier.avec.succes=Articles updated with success. |

**Le fichier messages\_fr.properties :**

|  |
| --- |
| #page articles.xhtml et edit.xhtml  titre=Exemple d'application web avec JSF+JPA+Wildfly  formulaire.article.designation=Désignation de l'article  formulaire.article.prix=Prix  formulaire.article.quantiteEnStock=Quantité en stock  #page articles.xhtml  ajouter.nouvel.article=Ajouter un nouvel article :  formulaire.bouton.ajouter=Ajouter  message.listeDesArticles=Liste des articles :  intitule.colonne.identifiant.listeDesArticles=Identifiant  intitule.colonne.designation.listeDesArticles=Désignation  intitule.colonne.prix.listeDesArticles=Prix  intitule.colonne.quantiteEnStock.listeDesArticles=Quantité en stock  intitule.lien.supprimer=Supprimer  intitule.lien.modifier=Modifier  #page edit.xhtml  modifier.article=Modifier un article :  formulaire.bouton.modifier=Modifier  #page login.xhtml  se.connecter=Autentification :  formulaire.login.username=Login  formulaire.login.password=Mot de passe  login.bouton.sautentifier=S'authentifier  logout=Fermer la session  #page panier.xhtml  msg.panier=Gestion du panier  #page accessDenied.xhtml  message.access.denied=Bonjour **{0}** : Vous n'avez pas le droit d'accéder à cette page. Merci de contacter l'administrateur.  lien.revenir.page.login=Revenir à la page d'authentification  #Validations messages  login.doesnt.exist=Le nom utilisateur inexstant.  password.incorrect=Mot de passe incorrect.  problem.user.role=Aucun rôle ne vous est attribué. Veuillez contacter votre 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 ajouté avec succès.  article.sypprime.avec.succes=Article supprimé avec succès.  article.modifier.avec.succes=Articles modifiés avec succès. |

**Le fichier messages\_ar.properties :**

|  |
| --- |
| #page articles.xhtml et edit.xhtml  titre=\u0628\u0631\u0646\u0627\u0645\u062C \u0645\u0639\u0644\u0648\u0645\u0627\u062A\u064A \u0645\u0646\u062C\u0632 \u0628\u061B JSF\u060CJPA \u0648 WILDFLY  formulaire.article.designation=\u0625\u0633\u0645 \u0627\u0644\u0645\u0646\u062A\u062C  formulaire.article.prix=\u0627\u0644\u0633\u0639\u0631  formulaire.article.quantiteEnStock=\u0643\u0645\u064A\u0629 \u0627\u0644\u0645\u0633\u062A\u0648\u062F\u0639  #page articles.xhtml  ajouter.nouvel.article=: \u0625\u0636\u0627\u0641\u0629 \u0645\u0646\u062A\u062C \u062C\u062F\u064A\u062F  formulaire.bouton.ajouter=\u0625\u0636\u0627\u0641\u0629  message.listeDesArticles=: \u0644\u0627\u0626\u062D\u0629 \u0627\u0644\u0645\u0646\u062A\u062C\u0627\u062A  intitule.colonne.identifiant.listeDesArticles=\u0645\u0631\u062C\u0639  intitule.colonne.designation.listeDesArticles=\u0625\u0633\u0645 \u0627\u0644\u0645\u0646\u062A\u062C  intitule.colonne.prix.listeDesArticles=\u0627\u0644\u0633\u0639\u0631  intitule.colonne.quantiteEnStock.listeDesArticles=\u0643\u0645\u064A\u0629 \u0627\u0644\u0645\u0633\u062A\u0648\u062F\u0639  intitule.lien.supprimer=\u0645\u0633\u062D  intitule.lien.modifier=\u062A\u063A\u064A\u064A\u0631  #page edit.xhtml  modifier.article=: \u062A\u063A\u064A\u064A\u0631 \u0645\u0646\u062A\u062C  formulaire.bouton.modifier=\u062A\u063A\u064A\u064A\u0631  #page login.xhtml(to be completed  se.connecter=: \u0625\u062A\u0628\u0627\u062A \u0627\u0644\u0647\u0648\u064A\u0629  formulaire.login.username=\u0625\u0633\u0645 \u0627\u0644\u0645\u0633\u062A\u0639\u0645\u0644  formulaire.login.password=\u0627\u0644\u0631\u0642\u0645 \u0627\u0644\u0633\u0631\u064A  login.bouton.sautentifier=\u0623\u062F\u062E\u0644  logout=\u062E\u0631\u0648\u062C  must.authentify=\u064A\u062C\u0628 \u0639\u0644\u064A\u0643 \u062A\u0633\u062C\u064A\u0644 \u0627\u0644\u062F\u062E\u0648\u0644 \u0628\u0627\u0633\u062A\u062E\u062F\u0627\u0645 \u062D\u0633\u0627\u0628 \u0627\u0644\u0645\u0633\u062A\u062E\u062F\u0645 \u0627\u0644\u062E\u0627\u0635 \u0628\u0643  #page panier.xhtml (to be completed)  msg.panier=\u0625\u062F\u0627\u0631\u0629 \u0633\u0644\u0629  #page accessDenied.xhtml  message.access.denied=\u0645\u0631\u062D\u0628\u0627 **{0}** : \u063A\u064A\u0631 \u0645\u0633\u0645\u0648\u062D \u0644\u0643 \u0628\u0627\u0644\u0648\u0635\u0648\u0644 \u0625\u0644\u0649 \u0647\u0630\u0647 \u0627\u0644\u0635\u0641\u062D\u0629. \u064A\u0631\u062C\u0649 \u0627\u0644\u0627\u062A\u0635\u0627\u0644 \u0628\u0627\u0644\u0645\u0633\u0624\u0648\u0644.  lien.revenir.page.login=\u0627\u0644\u0639\u0648\u062F\u0629 \u0625\u0644\u0649 \u0635\u0641\u062D\u0629 \u0627\u0644\u0645\u0635\u0627\u062F\u0642\u0629  #Validations messages  login.doesnt.exist=\u062A\u0633\u062C\u064A\u0644 \u0627\u0644\u062F\u062E\u0648\u0644 \u063A\u064A\u0631 \u0645\u0648\u062C\u0648\u062F  password.incorrect=\u0643\u0644\u0645\u0629 \u0627\u0644\u0645\u0631\u0648\u0631 \u063A\u064A\u0631 \u0635\u062D\u064A\u062D\u0629  problem.user.role=\u0644\u064A\u0633 \u0644\u062F\u064A\u0643 \u0623\u064A \u062F\u0648\u0631. \u064A\u0631\u062C\u0649 \u0627\u0644\u0627\u062A\u0635\u0627\u0644 \u0628\u0627\u0644\u0645\u0633\u0624\u0648\u0644 \u0627\u0644\u062E\u0627\u0635 \u0628\u0643  save.update=\u062D\u0641\u0638 \u0627\u0644\u062A\u062D\u062F\u064A\u062B  cancel.update=\u0625\u0644\u063A\u0627\u0621 \u0627\u0644\u062A\u062D\u062F\u064A\u062B  #Infos messages  article.ajoute.avec.succes=\u0627\u0644\u0645\u0627\u062F\u0629 \u0627\u0644\u0645\u0636\u0627\u0641\u0629 \u0628\u0646\u062C\u0627\u062D.  article.sypprime.avec.succes=\u0627\u0644\u0645\u0627\u062F\u0629 \u062D\u0630\u0641\u062A \u0628\u0646\u062C\u0627\u062D.  article.modifier.avec.succes=\u0627\u0644\u0645\u0642\u0627\u0644\u0627\u062A \u0627\u0644\u062A\u064A \u062A\u0645 \u062A\u062D\u0631\u064A\u0631\u0647\u0627 \u0628\u0646\u062C\u0627\u062D |

1. **Le fichier persistence.xml :**

Ce fichier doit être placé dans le dossier src/META-INF :

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <persistence version=*"2.1"*  xmlns=*"http://xmlns.jcp.org/xml/ns/persistence"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence\_2\_1.xsd"*>  <persistence-unit name=*"unite1"*>  <provider>org.hibernate.ejb.HibernatePersistence</provider>  <class>ma.gov.formation.cigma.service.modele.Article</class>  <class>ma.gov.formation.cigma.service.modele.User</class>  <class>ma.gov.formation.cigma.service.modele.UserRole</class>  <properties>  <!-- Classes persistantes -->  <property name=*"hibernate.show\_sql"* value=*"true"* />  <property name=*"hibernate.format\_sql"* value=*"true"* />  <property name=*"use\_sql\_comments"* value=*"true"* />  <!-- connexion JDBC -->  <property name=*"hibernate.connection.driver\_class"* value=*"com.mysql.jdbc.Driver"* />  <property name=*"hibernate.connection.url"* value=*"jdbc:mysql://localhost:3306/formation\_db"* />  <property name=*"hibernate.connection.username"* value=*"root"* />  <property name=*"hibernate.connection.password"* value=*"root"* />  <!-- création automatique du schéma -->  <property name=*"hibernate.hbm2ddl.auto"* value=*"update"* />  <!-- Dialecte -->  <property name=*"hibernate.dialect"* value=*"org.hibernate.dialect.MySQL5InnoDBDialect"* />  </properties>  </persistence-unit>  </persistence> |

1. **Les pages \*.xhtml**

**La page articles.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <f:event listener=*"#{loginBean.isAdmin}"* type=*"preRenderView"* />  <h:body>  <ui:include src=*"/header.xhtml"* />  <ui:include src=*"/article\_formulaire.xhtml"* />  <ui:include src=*"/list\_articles.xhtml"* />  <ui:include src=*"/footer.xhtml"* />  </h:body>  </f:view>  </html> |

**La page login.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <h:body>  <ui:include src=*"/header.xhtml"* />  <h:form id=*"LoginForm"*>  <h:panelGrid columns=*"2"* border=*"1"* cellpadding=*"10"* cellspacing=*"1"*>  <f:facet name=*"header"*>  <h:outputText value=*"#{msg['se.connecter']}"* />  </f:facet>  <h:outputLabel value=*"#{msg['formulaire.login.username']}"* />  <h:inputText id=*"userName"* value=*"#{loginBean.userVo.userName}"* size=*"20"* required=*"true"* immediate=*"true"*/>  <h:outputLabel value=*"#{msg['formulaire.login.password']}"* />  <h:inputSecret id=*"password"* value=*"#{loginBean.userVo.password}"* size=*"20"* required=*"true"* immediate=*"true"*/>  <f:facet name=*"footer"*>  <h:panelGroup style="display:*block*; text-align:*center*">  <h:commandButton value=*"#{msg['login.bouton.sautentifier']}"* action=*"#{loginBean.login}"*/>  </h:panelGroup>  </f:facet>  </h:panelGrid>  </h:form>  </h:body>  </f:view>  </html> |

**La page accessDenied.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <h:body>  <ui:include src=*"/header.xhtml"* />  <h:form>  <h3>  <h:outputFormat value=*"#{msg['message.access.denied']}"*>  <f:param value=*"#{user.userName}"*></f:param>  </h:outputFormat>  </h3>  <tr align=*"left"*>  <h:commandLink action=*"login"*  value=*"#{msg['lien.revenir.page.login']}"* />  </tr>  </h:form>  </h:body>  </f:view>  </html> |

**La page panier.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.panier}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <f:event listener=*"#{loginBean.isClient}"* type=*"preRenderView"* />  <h:body>  <ui:include src=*"/header.xhtml"* />  <ui:include src=*"/list\_articles.xhtml"* />  <ui:include src=*"/footer.xhtml"* />  </h:body>  </f:view>  </html> |

**La page article\_formulaire.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <h:body>  <h:form id=*"formulaire"*>  <h:panelGrid columns=*"2"* border=*"1"* cellpadding=*"10"* cellspacing=*"1"*>  <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:outputLabel value=*"#{msg['formulaire.article.prix']}"* />  <h:inputText id=*"price"* value=*"#{articleBean.articleVo.prix}"* required=*"true"* immediate=*"true"*/>  <h:outputLabel value=*"#{msg['formulaire.article.quantiteEnStock']}"* />  <h:inputText id=*"quantiteEnStock"* value=*"#{articleBean.articleVo.quantiteEnStock}"* required=*"true"* immediate=*"true"*/>  <f:facet name=*"footer"*>  <h:panelGroup style="display:*block*; text-align:*center*">  <h:commandButton value=*"#{msg['formulaire.bouton.ajouter']}"*  action=*"#{articleBean.addArticle}"*></h:commandButton>  </h:panelGroup>  </f:facet>  </h:panelGrid>  </h:form>  </h:body>  </f:view>  </html> |

**La page list\_articles.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <h:body>  <h:form id=*"list\_artciles"*>  <h2>#{msg['message.listeDesArticles']}</h2>  <h:dataTable value=*"#{articleBean.articlesVo}"* var=*"article"*  styleClass=*"articleTable"* headerClass=*"articleTableHeader"*  rowClasses=*"articleTableOddRow,articleTableEvenRow"*>  <h:column>  <f:facet name=*"header"*>#{msg['intitule.colonne.identifiant.listeDesArticles']}</f:facet>  <h:outputText value=*"#{article.id}"* />  </h:column>  <h:column>  <f:facet name=*"header"*>#{msg['intitule.colonne.designation.listeDesArticles']}</f:facet>  <h:outputText value=*"#{article.designation}"* rendered=*"#{not article.editable}"*/>  <h:inputText value=*"#{article.designation}"* rendered=*"#{article.editable}"*/>  </h:column>  <h:column>  <f:facet name=*"header"*>#{msg['intitule.colonne.prix.listeDesArticles']}</f:facet>  <h:outputText value=*"#{article.prix}"* rendered=*"#{not article.editable}"*/>  <h:inputText value=*"#{article.prix}"* rendered=*"#{article.editable}"*/>    </h:column>  <h:column>  <f:facet name=*"header"*>#{msg['intitule.colonne.quantiteEnStock.listeDesArticles']}</f:facet>  <h:outputText value=*"#{article.quantiteEnStock}"* rendered=*"#{not article.editable}"*/>  <h:inputText value=*"#{article.quantiteEnStock}"* rendered=*"#{article.editable}"*/>  </h:column>  <h:column>  <h:commandLink action=*"#{articleBean.removeArticle(article.id)}"*  value=*"#{msg['intitule.lien.supprimer']}"* />  </h:column>  <h:column>  <h:commandLink action=*"#{articleBean.editArticle(article.id)}"* value=*"#{msg['intitule.lien.modifier']}"* rendered=*"#{not article.editable}"* />  </h:column>  </h:dataTable>  <tr align=*"center"*>  <h:commandLink value=*"#{msg['save.update']}"* action=*"#{articleBean.saveUpdate}"* rendered=*"#{articleBean.editable}"*/>  </tr>  <tr align=*"center"*>  <h:commandLink value=*"#{msg['cancel.update']}"* action=*"#{articleBean.cancelUpdate}"* rendered=*"#{articleBean.editable}"*/>  </tr>  </h:form>  </h:body>  </f:view>  </html> |

**La page header.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <h:body>  <h:form id=*"langues"*>  <h:commandLink action=*"#{langue.changeLanguage('ar')}"*>  <h:graphicImage value=*"/resources/images/ar.png"* />  </h:commandLink>    <h:commandLink action=*"#{langue.changeLanguage('fr')}"*>  <h:graphicImage value=*"/resources/images/fr.png"* />  </h:commandLink>    <h:commandLink action=*"#{langue.changeLanguage('en')}"*>  <h:graphicImage value=*"/resources/images/en.png"* />  </h:commandLink>  </h:form>  <h:form id=*"messages"*>  <h:messages infoStyle=*"color:green"* errorStyle=*"color:red"*/>  </h:form>  </h:body>  </f:view>  </html> |

**La page footer.xhtml :**

|  |
| --- |
| <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:h=*"http://xmlns.jcp.org/jsf/html"*>  <f:view locale=*"#{langue.locale}"*>  <h:head>  <title>#{msg.titre}</title>  <h:outputStylesheet library=*"css"* name=*"styles.css"* />  </h:head>  <h:body>  <h:form>  <h:commandLink action=*"#{loginBean.logout}"* value=*"#{msg.logout}"* />  </h:form>  </h:body>  </f:view>  </html> |

1. **La feuille styles.css**

**Ce fichier doit être mis dans le dossier : WebContent/resources/css/.**

|  |
| --- |
| *.articleTable* {  border-collapse:*collapse*;  border:*1px* *solid* *#000000*;  }  *.articleTableHeader* {  text-align:*center*;  background:*none* *repeat* *scroll* *00#B5B5B5*;  border-bottom:*1px* *solid* *#000000*;  padding:*2px*;  }  *.articleTableOddRow* {  text-align:*center*;  background:*none* *repeat* *scroll* *00#FFFFFFF*;  }  *.articleTableEvenRow* {  text-align:*center*;  background:*none* *repeat* *scroll* *00#D3D3D3*;  }  *.centered* {  margin: *0px* *auto*;  text-align: *center*;  }    *.left* {  float: *left*;  } |

1. **Les images**

Copier les images ar.png, fr.png et en.png dans le dossier : WebContent/resources/images/.

1. **Le fichier faces-config.xml**

|  |
| --- |
| <?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>  <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. **Le fchier 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>login.xhtml</welcome-file>  </welcome-file-list>  <servlet>  <servlet-name>Faces Servlet</servlet-name>  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>Faces Servlet</servlet-name>  <url-pattern>\*.xhtml</url-pattern>  </servlet-mapping>  </web-app> |

1. **Les classes de test**

La classe test.user.Test :

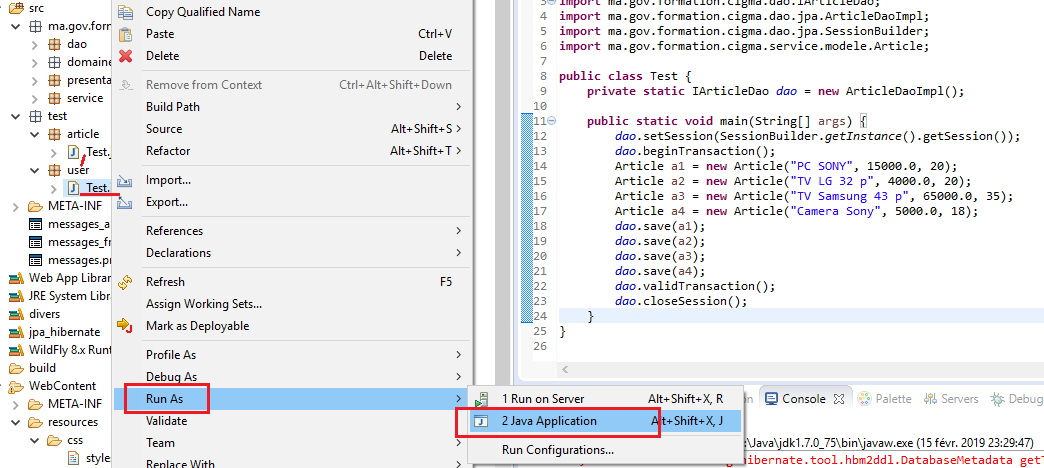
|  |
| --- |
| **package** test.user;  **import** ma.gov.formation.cigma.dao.IUserDao;  **import** ma.gov.formation.cigma.dao.jpa.SessionBuilder;  **import** ma.gov.formation.cigma.dao.jpa.UserDaoImpl;  **import** ma.gov.formation.cigma.presentation.utils.Constants;  **import** ma.gov.formation.cigma.service.modele.User;  **import** ma.gov.formation.cigma.service.modele.UserRole;  **public** **class** Test {  **private** **static** IUserDao *dao* = **new** UserDaoImpl();  **public** **static** **void** main(String[] args) {  *dao*.setSession(SessionBuilder.*getInstance*().getSession());  *dao*.beginTransaction();  UserRole adminRole1=**new** UserRole(Constants.***ADMIN\_ROLE***,Constants.***ADMIN\_ROLE\_DESCRIPTION***);  UserRole adminRole2=**new** UserRole(Constants.***ADMIN\_ROLE***,Constants.***ADMIN\_ROLE\_DESCRIPTION***);  UserRole clientRole1=**new** UserRole(Constants.***CLIENT\_ROLE***,Constants.***CLIENT\_ROLE\_DESCRIPTION***);  UserRole clientRole2=**new** UserRole(Constants.***CLIENT\_ROLE***,Constants.***CLIENT\_ROLE\_DESCRIPTION***);    User user1=**new** User("admin","admin");user1.addUserRole(adminRole1);  User user2=**new** User("client","client");user2.addUserRole(clientRole1);  User user3=**new** User("superadmin","superadmin");user3.addUserRole(adminRole2);user3.addUserRole(clientRole2);  *dao*.save(user1);  *dao*.save(user2);  *dao*.save(user3);  *dao*.validTransaction();  *dao*.closeSession();  }  } |

La classe test.article.Test :

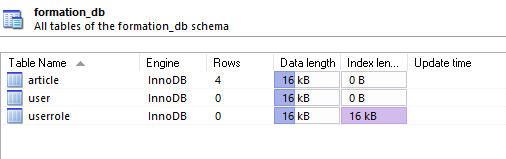
|  |
| --- |
| **package** test.article;  **import** ma.gov.formation.cigma.dao.IArticleDao;  **import** ma.gov.formation.cigma.dao.jpa.ArticleDaoImpl;  **import** ma.gov.formation.cigma.dao.jpa.SessionBuilder;  **import** ma.gov.formation.cigma.service.modele.Article;  **public** **class** Test {  **private** **static** IArticleDao *dao* = **new** ArticleDaoImpl();  **public** **static** **void** main(String[] args) {  *dao*.setSession(SessionBuilder.*getInstance*().getSession());  *dao*.beginTransaction();  Article a1 = **new** Article("PC SONY", 15000.0, 20);  Article a2 = **new** Article("TV LG 32 p", 4000.0, 20);  Article a3 = **new** Article("TV Samsung 43 p", 65000.0, 35);  Article a4 = **new** Article("Camera Sony", 5000.0, 18);  *dao*.save(a1);  *dao*.save(a2);  *dao*.save(a3);  *dao*.save(a4);  *dao*.validTransaction();  *dao*.closeSession();  }  } |

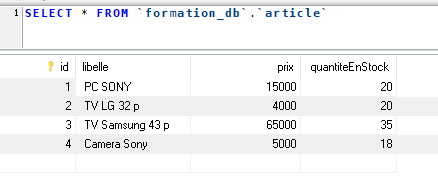
# Les tests :

1) Lancer la méthode main de la classe test.article.Test comme illustré ci-dessous :

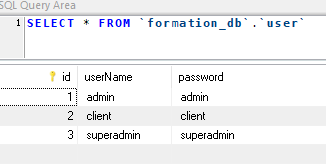


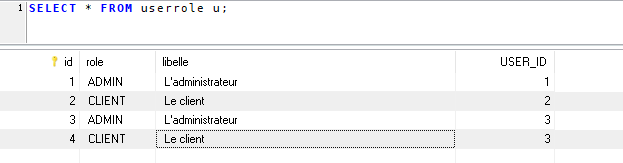
Vérifier que les 03 tables ARTICLE, USER et USERROLE ont été bien créés et vérifier que les 04articles ont été bien ajoutés au niveau de la table ARTICLE :





2) Lancer la méthode main de la classe test.user.Test et vérifies que les lignes ont été ajoutées au niveau des tables USER et USERROLE :





**Remarque :**

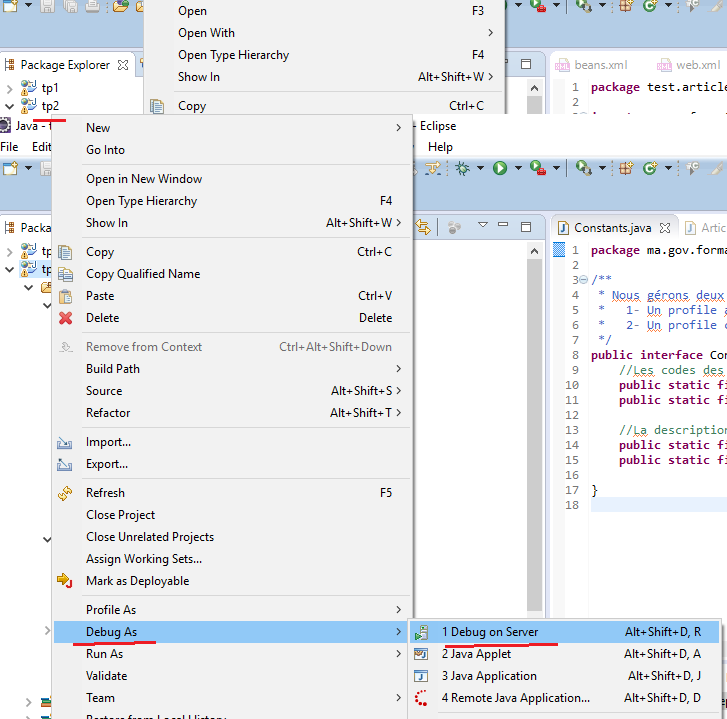
Le tableau ci-après récapitule la liste des utilisateurs avec leurs profiles respectifs :

|  |  |  |
| --- | --- | --- |
| Utilisateur | Mot de passe | Rôles |
| admin | admin | ADMIN |
| client | client | CLIENT |
| superadmin | superadmin | ADMIN, CLIENT |

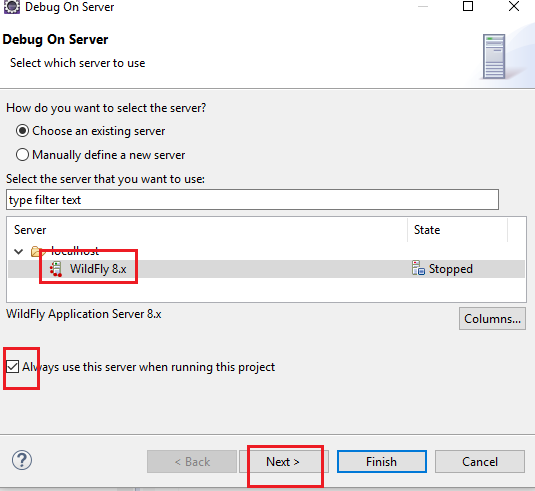
Vous pouvez créer vos propres utilisateurs et leurs affecter soit le rôle ADMIN ou bien CLIENT.

**3) Tester l'application web :**

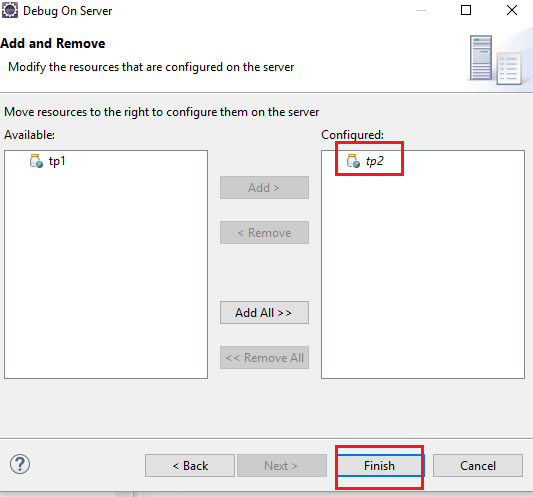
Exécuter votre application TP2 comme illustré ci-dessous :



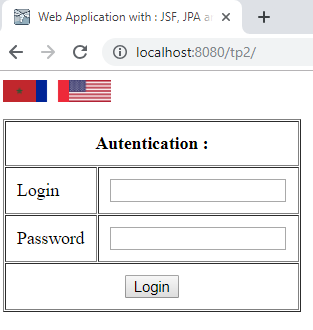
Cliquer sur Debug on Server :



Sélectionner WildFly et cliquer ensuite sur Next > :

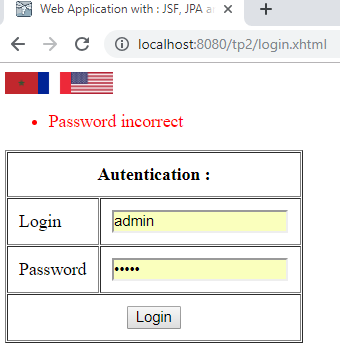


Cliquer ensuite sur Finish. La page suivante sera affichée:

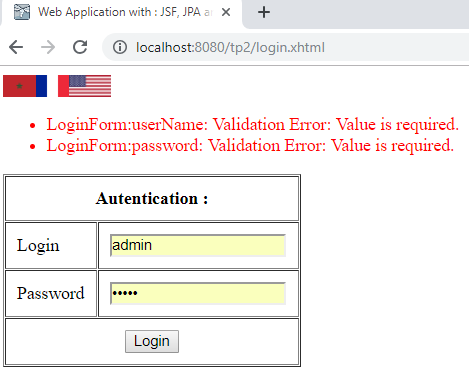


**Cas de test n°1 : L'utilisateur admin ayant le rôle ADMIN**

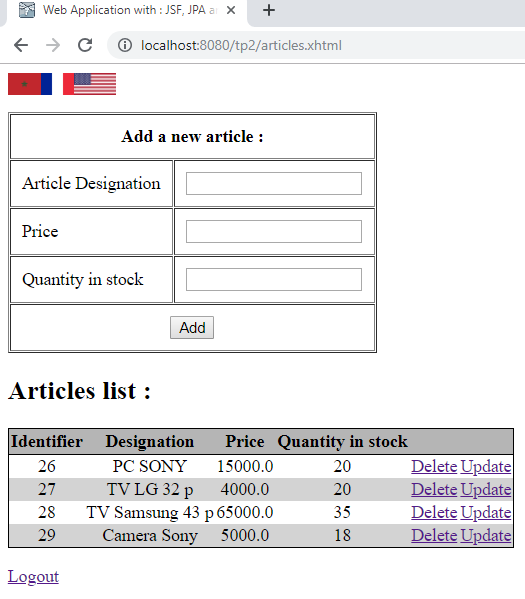
1- Entrer un mot de passe erroné, le message suivant sera affiché :

****

2- Ne rien saisir, le message suivant sera affiché :

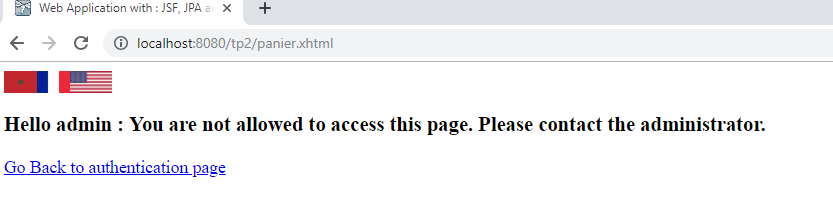
****

3- Enter le bon mot de passe, la page suivante est affichée :



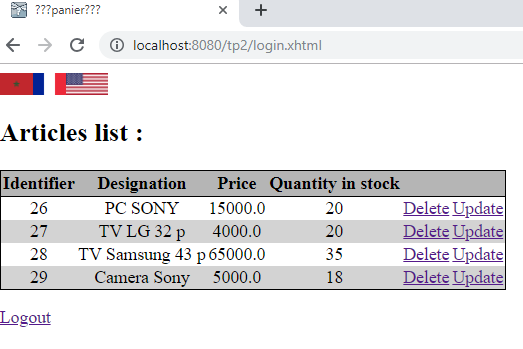
4- Pour fermer la session cliquer sur le lien "Fermer la session".

5- Accéder à nouveau à l'application avec le compte admin et ensuite exécuter le lien : http://localhost:8080/tp2/panier.xhtml. Vérifier que l'application redirige la réponse vers la page accessDenied.xhtml comme le montre la page suivante :



**- Cas de test n°2 : L'utilisateur client ayant le rôle CLIENT**

1- Se connecter avec un compte client, la page suivante sera affichée :



2- Exécuter le lien http://localhost:8080/tp2/articles.xhtml, la page suivante est affichée :

**- Cas de test n°3 : L'utilisateur superadmin ayant le rôle ADMIN et ROLE :**

Vérifier que le compte superadmin a le droit d'accéder à toutes les pages.