

---

# JAVA EE : COMPTE-RENDU DU PROJET

---

**Numéro de Référence**

**#UNIVAMU001**

(Document de 7 pages)

---

## Résumé

---

Ce document est un rapport de ce qui a été fait lors du projet de JEE portant sur la gestion d'annuaire de personnes enseigné par M. Jean-Luc Massat. En outre, ce TP traitera dans un premier temps la mise en place de la couche de stockage puis de la couche de présentation à travers l'utilisation de la technologie Java EE .

---

## Mots Clés

---

Java EE, J2EE, DAO, JDBC, JSP, Servlet, application WEB, JSTL, xml, XHTML, Eclipse.

Aix-Marseille Université  
**Faculté des Sciences**  
163 Avenue de Luminy  
Case 901 13288 MARSEILLE CEDEX 13

## SECTION DES REDACTEURS

Nom	Prénom	Contribution
Gairoard	Lionel	Rédaction du rapport
Pachy	Ravi	Rédaction du rapport

## CONTACTS

Nom	Prénom	Email	Fonction
Gairoard	Lionel	<a href="mailto:lionel.gairoard@gmail.com">lionel.gairoard@gmail.com</a>	Etudiant
Pachy	Ravi	<a href="mailto:pachy.ravi@gmail.com">pachy.ravi@gmail.com</a>	Étudiant

## HISTORIQUE DES MODIFICATIONS

Modifications	Date	Version	Approbateur de la diffusion
Rédaction du rapport	28/01/14	1.0	Gairoard Lionel, Pachy Ravi

## TABLE DES MATIERES

<b>1. Introduction .....</b>	<b>4</b>
<b>2. Partie I : Couche de stockage (DAO / JDBC) .....</b>	<b>4</b>
2.1. Conception de la base de données.....	4
2.2. Conception du JavaBean.....	4
2.3. Conception de la DAO.....	4
2.4. Conception des tests unitaires.....	4
2.5. Résultats des tests.....	5
<b>3. Partie II : Couche de présentation (JSP / Servlet) .....</b>	<b>6</b>
<b>4. Conception de la servlet de services.....</b>	<b>6</b>
<b>5. Conception des pages JSP et CSS .....</b>	<b>6</b>
5.1. Page <i>index.jsp</i> .....	6
5.2. Page <i>directory.jsp</i> .....	6
5.3. Page <i>details.jsp</i> .....	6
5.4. Page <i>backup.jsp</i> .....	6
5.5. Page <i>edition.jsp</i> .....	7
5.6. Page <i>edition.jsp</i> .....	7
5.7. Page <i>backup.jsp</i> .....	7
5.8. Page <i>form.css, table.css</i> .....	7
<b>6. Conclusion .....</b>	<b>7</b>

## 1. INTRODUCTION

Au sein de notre parcours de formation en Master 2 Informatique à la faculté des sciences de Luminy, il nous a été demandé de réaliser un projet permettant la gestion d'un annuaire de personnes à l'aide des technologies Java EE. Ce projet s'inscrit dans le cours d'architecture Java EE enseigné par M. Jean-Luc Massat, enseignant-chercheur au Département d'informatique et membre de l'équipe Combinatoire et Recherche Opérationnelle du Laboratoire d'informatique Fondamentale de Marseille.

Le projet est découpé en deux parties : La première concerne la mise en œuvre d'une couche de service d'accès aux données (Data Access Object). La deuxième permet la création d'une application WEB permettant de gérer efficacement les actions d'un client afin de modifier un annuaire de personnes.

## 2. PARTIE I : COUCHE DE STOCKAGE (DAO / JDBC)

Dans cette section nous allons expliquer comment nous avons procédé pour réaliser notre DAO.

### 2.1. Conception de la base de données

Afin de pouvoir gérer un annuaire de personnes, il convient tout d'abord de définir au sein d'une base de données relationnelle notre représentation d'une personne.

Nous avons opté pour le SGBDR PostgreSQL en raison de ses avantages par rapport à ses concurrents (respect de la norme SQL2003, opensource, communauté active...).

Notre personne sera représentée par un identifiant (entier), un nom, prénom, adresse électronique, site WEB (facultatif), date de naissance et d'un mot de passe.

Le code SQL relatif à la création de la table *Personne* est donné en annexe.

### 2.2. Conception du JavaBean

Notre classe valeur est représentée par le JavaBean *Person.java* fourni en annexe.

Notons que l'on prendra soin d'implémenter l'interface *Serializable* dans notre JavaBean afin de pouvoir sauvegarder son état au cours de l'utilisation de l'application.

### 2.3. Conception de la DAO

Notre service d'accès aux données (DAO) est constitué d'une interface *DirectoryDAO.java* définissant les opérations nécessaire à la gestion de nos données et de son implémentation *DirectoryDAOImp.java*.

Notons que pour utiliser la technologie JDBC nous avons dû télécharger le driver à l'adresse <http://jdbc.postgresql.org/download.html> et d'inclure le *.jar* associé au sein du "Java Build Path" de notre application.

### 2.4. Conception des tests unitaires

Afin de garantir la fiabilité de notre programme, nous avons mis en place une classe de test à l'aide de l'outil *JUnit* qui nous a permis de valider les méthodes offertes par notre DAO au sein de la classe java *DirectoryJDBCTest.java*.

Détail des tests :

- **setUp()** : Exécutée avant le début des tests, cette méthode assigne les deux beans Person de travail.
- **newConnection()** : Teste la création de la connexion par l'implémentation de la DAO.
- **addPerson1()** : Teste l'ajout de la personne 1 dans la BDD.
- **addPerson2()** : Teste l'ajout de la personne 2 dans la BDD.
- **findPerson()** : Teste la récupération de la personne 1 après insertion.
- **findAllPersons()** : Teste la récupération de toutes les personnes après l'ajout de la personne 1 et de la personne 2.
- **updatePerson()** : Teste la mise à jour de la personne 1 après modification de son nom et de son prénom.
- **deletePerson1()** : Teste la suppression de la personne 1 après l'avoir ajoutée dans la BDD.

## 2.5. Résultats des tests

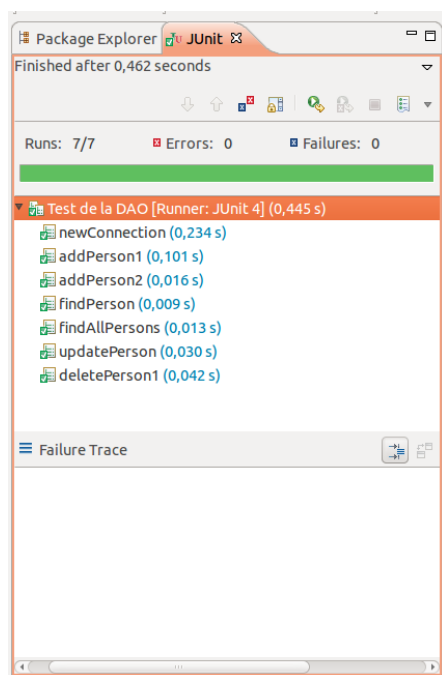


Figure 1 Résultats des tests

Coverage			
DirectoryJDBCTest (1) (2 Févr. 2014 21:51:31)			
Element	Coverage	Covered Instructions	Missed Instructions
JEE-Projet	58,8 %	798	559
src	58,8 %	798	559
fr.jee.projet.servlet	0,0 %	0	496
fr.jee.projet.dao	84,8 %	352	63
DirectoryDAOImp.java	84,8 %	352	63
fr.jee.projet.dao.test	100,0 %	373	0
DirectoryJDBCTest.java	100,0 %	373	0
fr.jee.projet.db	100,0 %	73	0
Person.java	100,0 %	73	0

Figure 2 Rapport de couverture des testss

### 3. PARTIE II : COUCHE DE PRESENTATION (JSP / SERVLET)

Dans cette partie nous allons décrire le contrôleur qui assure le traitement et la validation des requêtes du client à l'aide d'une *servlet* centrale mais aussi les vues permettant l'affichage des résultats à partir des données par la création de page *JSP*.

### 4. CONCEPTION DE LA SERVLET DE SERVICES

Par souci de simplification, nous avons choisi d'intégrer les services de notre application au sein d'une unique *servlet* *Controler.java* jointe en annexe. Celle-ci redirige les requêtes clientes vers l'une des cinq pages *JSP* : *index.jsp*, *directory.jsp*, *details.jsp*, *edition.jsp* et *backup.jsp* en fonction de l'action demandée en entrée.

Le traitement se fait côté *servlet* et les *JSP* servent de vue.

### 5. CONCEPTION DES PAGES JSP ET CSS

Comme nous l'avons vu, notre contrôleur s'architecture autour de quatre pages *JSP*. Voyons en détails leur composition.

#### 5.1. Page *index.jsp*

*Index.jsp* est la page de bienvenue lors du lancement de l'application. Elle permet la connexion de l'utilisateur responsable de la gestion de l'annuaire de personne. Pour ce faire, il devra entrer les informations d'authentification telles que son identifiant (adresse mail) et son mot de passe de connexion.

Une fois ces derniers renseignés, il peut avoir accès à la liste de personnes présente dans l'annuaire et effectuer s'il le désire des modifications.

#### 5.2. Page *directory.jsp*

Cette page permet de lister les noms des personnes présentes dans l'annuaire. Le gestionnaire peut alors cliquer sur une personne de l'annuaire pour afficher en détails ses renseignements et procéder à son édition.

#### 5.3. Page *details.jsp*

La page *details.jsp* affiche la fiche détaillée de la personne sélectionnée dans l'annuaire de personnes.

#### 5.4. Page *backup.jsp*

Cette page affiche une confirmation ou un message d'erreur après avoir tenté d'ajouter ou de modifier une personne dans la base de données.

Une erreur est affichée dans le cas où un problème est survenu durant l'opération réalisée par la DAO.



### 5.5. Page *edition.jsp*

Cette page sert à générer un formulaire d'édition/ajout d'une personne. La différence entre ces deux opérations est faite grâce à l'utilisation d'un champ input de type « hidden » contenant l'id de la personne en cours de modification. Cet id vaut -1 dans le cas où la personne n'existe pas (i.e. dans le cas où c'est un ajout et non une modification).

### 5.6. Page *edition.jsp*

L'édition permet de mettre un utilisateur en session et construit le formulaire permettant son enregistre ou sa mis

### 5.7. Page *backup.jsp*

Elle permet l'enregistrement / l'update d'une personne (après vérification) au sein de la base de donnée d'un groupe de personne grâce au formulaire fourni lors de l'édition.

### 5.8. Page *form.css, table.css*

Ces fichiers .css permettent la mise en forme de nos pages .jsp. On retrouve deux fichiers principaux :

- *form.css* qui gère la présentation de la page d'accueil, de l'édition / sauvegarde et du descriptif détaillée d'une personne.
- *table.css* qui permet l'affichage sous forme de tableau de notre annuaire de personne.

## 6. CONCLUSION

Ce projet nous a permis de nous familiariser avec les technologies JEE. Nous avons aussi vu qu'il était intéressant de séparer notre application en module afin de gérer au mieux les interactions Client / Serveur.