

## JakartaEE – Intégration Bootstrap et javascript (1)

Nous allons créer une nouvelle application permettant de gérer le personnel d'une entreprise de travaux publics. Cette application nécessite :

- une authentification,
- puis propose :
  - de créer un salarié,
  - de lister les salariés
  - et de permettre en début de chaque semaine d'affecter le personnel sur chantier, au bureau ou en absence.

### I. Authentification

Seul le chef de chantier peut accéder à l'application. Il doit être présent dans la table *utilisateur* de la base de données *bdpersonnel* (script de création disponible sur le Drive).

La saisie d'un login et d'un mot de passe contrôlés en base de données, permettra l'accès aux fonctionnalités de l'application.

Travail à faire :	
1.	Afin de recenser les classes nécessaires et de bien comprendre leurs interactions, réaliser un diagramme de séquence.
<b>Le travail suivant est à faire sur poste : utiliser un dépôt gitea privé nommé PersonnelTP et donner les droits à l'utilisateur delphine.thevenot pour pouvoir l'utiliser</b>	
2.	Récupérer depuis le Drive : <ul style="list-style-type: none"> <li>• Le script de création de la base de données</li> <li>• Le fichier compressé personnelTP1.zip</li> </ul>
3.	Après démarrage des serveurs web Apache et MySQL, exécuter phpMyadmin et exécuter le script de création de la base de données. Démarrer NetBeans et créer un nouveau projet javaWeb nommé PersonnelTP Tester Décompresser personnelTP1.zip Recopier le dossier jspf dans WEB-INF Ajouter les bibliothèques jstl et mysql
4.	Dans WEB-INF, créer la jsp authentifVue. Ajouter les include nécessaires et créer le formulaire d'identification. Cf exemples avec utilisation de Bootstrap 5 : <a href="https://getbootstrap.com/docs/5.0/forms/overview/">https://getbootstrap.com/docs/5.0/forms/overview/</a> , icones : <a href="https://icons.getbootstrap.com/">https://icons.getbootstrap.com/</a>
5.	Créer le package controleur dans Source Packages Créer la servlet AuthentifServlet, sans oublier de cocher Add information to deployment descriptor. Au niveau de l'URL pattern, préciser /authentification. Tester, url : <a href="http://localhost:8080/PersonnelTP/authentification">http://localhost:8080/PersonnelTP/authentification</a> Modifier la méthode doGet de façon à afficher la jsp authentifJSP.jsp Tester l'affichage du formulaire
6.	Créer le package modelesBean dans Source Packages. Créer le bean Utilisateur

7.	<p>Créer le package bdd dans Source Packages.  Recopier la classe Connexion(celle du projet précédent)  Créer la classe utilisateurMysql.  Dans cette classe, créer</p> <ul style="list-style-type: none"> <li>• le constructeur, qui doit faire appel à la classe Connexion</li> <li>• la méthode readAuthentif qui devra vérifier s'il existe un utilisateur disposant du login et du mot de passe passé en paramètres.</li> </ul> <p>Ne pas oublier d'utiliser une requête préparée :</p> <pre>public Utilisateur readAuthentif(String login, String mdp) {     Utilisateur util = null;     try {         PreparedStatement prepStmt = null;         String sql = "SELECT * FROM utilisateur WHERE login=? AND mdp=? ";         prepStmt = theConnection.prepareStatement(sql);         prepStmt.setString(1, login);         prepStmt.setString(2, mdp);         ResultSet result = prepStmt.executeQuery();         if (result.next()) {             util = new Utilisateur(result.getInt(1), result.getString(2),                 result.getString(3), result.getString(4), result.getString(5));         }         prepStmt.close();     } catch (SQLException ex) {         System.out.println("SQLException : " + ex.getMessage());         System.out.println("SQLState : " + ex.getSQLState());         System.out.println("Code erreur : " + ex.getErrorCode());     }     return util; }</pre>
8.	<p>Créer le package formulaires dans Sources Package  Y ajouter la classe AuthentifForm.java  Cette classe comprend</p> <ul style="list-style-type: none"> <li>• 1 attribut : <ul style="list-style-type: none"> <li>o HttpServletRequest request;</li> </ul> </li> <li>• Un constructeur qui valorisera l'attribut request</li> <li>• La méthode ctrlAuthentif qui retournera un booléen. Cette méthode <ul style="list-style-type: none"> <li>o Récupère les login et mot de passe</li> <li>o Chiffre le mot de passe</li> <li>o Crée une instance de UtilisateurMysql</li> <li>o Appelle la méthode readAuthentif.</li> </ul> </li> </ul> <p>Si le contrôle est réussi, l'utilisateur sera mis en session, Sinon, l'attribut message sera renseigné  Retourner le booléen adéquat.</p>
9.	<p>Modification de la servlet authentifServlet  Mettre à jour la méthode doPost :</p> <ul style="list-style-type: none"> <li>• Créer une instance de de AuthentifForm</li> <li>• Faire appel à la méthode ctrlAuthentif</li> <li>• Si l'authentification n'a pas réussi, rediriger l'utilisateur vers la jsp uthentifJSP.jsp</li> <li>• Sinon, rediriger l'utilisateur vers la jsp menuJSP.jsp qui sera à créer par la suite.</li> </ul>

## II. Poursuite de l'application

Si l'utilisateur s'authentifie avec succès, il faudrait lui proposer une page avec 3 choix :

- Création d'un nouveau salarié
- Liste des salariés
- Affectation des salariés

### Saisie d'un nouveau salarié

Si l'utilisateur choisit la création d'un nouveau salarié, un formulaire doit permettre de saisir son nom, son prénom et sa fonction. Après validation un message doit indiquer que le salarié à bien été créé.

Développer cette fonctionnalité en créant les éléments nécessaires.