
JAVAEE

COMPTE RENDU

ATELIER 1

Réalisé par : *EL FATHI Zakaria*

Encadré par : *Pr. BAKKOURI Ibtissam*



SOMMAIRE

INTRODUCTION	3
PARTIE 1 : CREER UN ETUDIANT.....	3
PARTIE 2 : CREER UN ETUDIANT.....	10
CONCLUSION:	17

INTRODUCTION

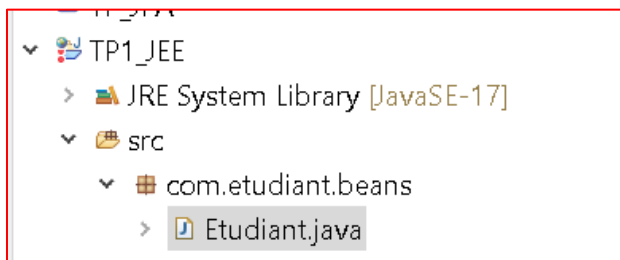
Le but de ce TP est de créer une application de gestion de l'inscription des étudiants en utilisant le modèle MVC (Model-View-Controller). La partie 1 de ce TP consiste à créer une application où l'utilisateur peut créer un étudiant en saisissant les données nécessaires depuis un formulaire et visualiser la fiche étudiant en résultant. Dans la partie 2, nous allons ajouter un formulaire nommé inscription qui permet également à l'utilisateur de remplir les informations d'inscription d'un étudiant donné.

Pour atteindre ces objectifs, nous allons utiliser les technologies suivantes : Eclipse, Tomcat, JSP, Servlet, JEE, HTML, CSS.

PARTIE 1 : CREER UN ETUDIANT

Dans cette partie, nous allons créer un bean pour représenter un étudiant, une servlet pour récupérer les données du formulaire et les transmettre à une JSP pour afficher la fiche étudiant créé.

Bean pour représenter un étudiant :



Un bean est une classe Java qui contient des propriétés (variables) et des méthodes (fonctions) pour accéder et manipuler ces propriétés. Nous allons créer une classe Java nommée "Etudiant" pour représenter un étudiant.

Voici le code de la classe "Etudiant" :

```
package com.etudiant.beans;

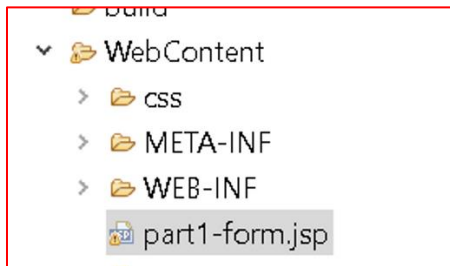
public class Etudiant {
    // Définition des propriétés de l'étudiant
    private static String firstName; // prénom
    private static String lastName; // nom de famille
    private static String email; // adresse e-mail
    private static String cne; // Code National de l'Etudiant
    private static String dNaiss; // date de naissance
    private static String tele; // numéro de téléphone

    // Accesseur pour le prénom
```

```
public String getFirstName() {
    return firstName;}
// Mutateur pour le prénom
public void setFirstName(String firstName) {
    this.firstName = firstName;}
// Accesseur pour le nom de famille
public String getLastName() {
    return lastName;}
// Mutateur pour le nom de famille
public void setLastName(String lastName) {
    Etudiant.lastName = lastName;
    // Utilisation de "Etudiant." pour accéder à la variable statique lastName
    // plutôt qu'à la variable d'instance "this.lastName"}
// Accesseur pour l'adresse e-mail
public String getEmail() {
    return email;}
// Mutateur pour l'adresse e-mail
public void setEmail(String email) {
    Etudiant.email = email;}
// Accesseur pour la date de naissance
public String getDNaiss() {
    return dNaiss;}
// Mutateur pour la date de naissance
public void setdNaiss(String dNaiss) {
    this.dNaiss = dNaiss;}
// Mutateur pour le Code National de l'Etudiant
public void setCne(String cne) {
    this.cne = cne;}
// Accesseur pour le Code National de l'Etudiant
public String getCne() {
    return cne;}
// Accesseur pour le numéro de téléphone
public String getTele() {
    return tele;}
// Mutateur pour le numéro de téléphone
public void setTele(String tele) {
    this.tele = tele;
}
}
```

Bien qu'il existe différentes façons de collecter des données à partir d'un utilisateur, l'utilisation d'un formulaire est l'une des plus courantes sur le web. Dans l'exemple suivant, nous avons un formulaire qui permet à un étudiant de s'inscrire en saisissant différentes informations telles que son nom, son prénom, son adresse e-mail, son CNE, sa date de naissance et son numéro de téléphone. Une fois que l'utilisateur a rempli le formulaire et cliqué sur le bouton "Envoyer", les données sont envoyées à une servlet nommée "part1Servlet". Cette servlet sera responsable de traiter les données saisies par l'utilisateur, selon la logique métier de l'application. Le formulaire est donc un élément clé de l'interaction entre l'utilisateur et l'application web, car il permet à l'utilisateur de fournir

des informations qui peuvent être utilisées pour personnaliser l'expérience utilisateur et offrir des fonctionnalités supplémentaires.



```
<html lang="en">
<head>
<title>formulaire Etudiant</title>
<meta charset="utf-8">
<meta name="viewport"
  content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="css/style.css">
</head>
<body>
  <!-- Login box container -->
  <div class="login-box">
    <h2>Inscription:</h2>
    <!-- Form for student registration -->
    <form method="post" action="part1Servlet">
      <div class="input-group">
        <!-- First name input -->
        <div class="user-box">
          <input type="text" class="form-control" id="firstName"
            name="firstName" placeholder="votre prenom">
        </div>
        <!-- Last name input -->
        <div class="user-box">
          <input type="text" class="form-control" id="lastName"
            name="lastName" placeholder="votre nom">
        </div>
        <!-- Email input -->
        <div class="user-box">
          <input type="email" class="form-control" id="email" name="email"
            placeholder="votre email">
        </div>
        <!-- CNE input -->
        <div class="user-box">
          <input type="text" class="form-control" id="cne" name="cne"
            placeholder="votre CNE">
        </div>
        <!-- Date of birth input -->
```

```
<label class="user-box">Date de naissance:</label>
<div class="user-box">
  <input type="date" id="dNaiss" name="dNaiss">
</div>
<!-- Telephone input -->
<div class="user-box">
  <input type="tele" class="form-control" id="tele" name="tele"
    placeholder="votre numero de telephone">
</div>
<!-- Submit button -->
<input type="submit" value="Envoyer!">
</div>
</form> </div>
</body>
</html>
```

ensuite on definie une Servlet pour récupérer les données du formulaire :



Le code est une classe Java nommée "part1Servlet" située dans le package "com.part1.servlets". Elle étend la classe HttpServlet, qui fournit des méthodes pour la gestion des requêtes HTTP. La classe est annotée avec l'annotation "@WebServlet" qui spécifie le chemin d'accès à cette servlet.

La méthode doPost() est surchargée pour traiter les requêtes POST. Elle récupère les paramètres du formulaire envoyé depuis la page "part1-form.jsp", tels que le prénom, le nom, l'e-mail, le CNE, la date de naissance et le téléphone. Ensuite, elle crée une nouvelle instance de la classe "Etudiant" et utilise ses setters pour remplir ses propriétés avec les données récupérées du formulaire.

Ensuite, elle stocke l'objet "Etudiant" dans l'objet requête en utilisant la méthode setAttribute(). Enfin, elle transmet la requête et la réponse à la page "part1-response.jsp" en utilisant la méthode RequestDispatcher().

Voici le code de la servlet "part1Servlet" :

```
package com.part1.servlets;
@WebServlet("/part1Servlet")
public class part1Servlet extends HttpServlet {
  private static final long serialVersionUID = 1L;
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Retrieve the first name and last name and email from myForm.jsp
    String firstName = request.getParameter("firstName");
    String lastName = request.getParameter("lastName");
    String email = request.getParameter("email");
    String cne = request.getParameter("cne");
    String tele = request.getParameter("tele");
    String dNaiss = request.getParameter("dNaiss");

    // Create a new instance of the Etudiant and populate its properties
    Etudiant Etudiant = new Etudiant();
    Etudiant.setFirstName(firstName);
    Etudiant.setLastName(lastName);
    Etudiant.setEmail(email);
    Etudiant.setTele(tele);
    Etudiant.setdNaiss(dNaiss);
    Etudiant.setCne(cne);
    // Store the Etudiant in the request object
    request.setAttribute("Etudiant", Etudiant);
    // Forward the request to the myResponse.jsp page
    request.getRequestDispatcher("/part1-response.jsp").forward(request, response);
}
}
```

Ensuite les données du formulaire doivent s'afficher dans la page part1-response.JSP suivante:

```
<!DOCTYPE html>
<%@ page import="com.etudiant.beans.Etudiant"%>
<!-- Importe la classe Etudiant -->

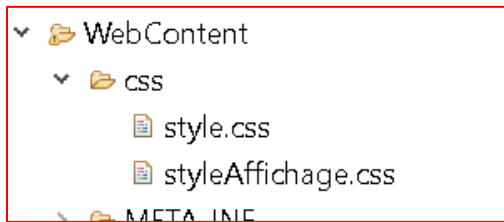
<html>
<head>
<link rel="stylesheet" href="css/styleAffichage.css">
<!-- Importe la feuille de style CSS -->
<title>my Response</title>
<!-- Titre de la page HTML -->
</head>
<body>
    <div class="login-box">
        <!-- Définit une boîte pour le contenu -->
        <div class="input-group">
            <div class="user-box">
                <h1>
                    <strong>PRENOM:</strong>
                    <%= (Etudiant) request.getAttribute("Etudiant").getFirstName() %>
                </h1>
            </div>
        </div>
    </div>
</body>
</html>
```

```

    </h1>
    <!-- Récupère et affiche le prénom de l'étudiant -->
</div>
<div class="user-box">
    <h1>
        <strong>NOM:</strong>
        <%=((Etudiant) request.getAttribute("Etudiant")).getLastName()%></h1>
    <!-- Récupère et affiche le nom de l'étudiant -->
</div>
<div class="user-box">
    <h1>
        <strong>EMAIL:</strong>
        <%=((Etudiant) request.getAttribute("Etudiant")).getEmail()%></h1>
    <!-- Récupère et affiche l'email de l'étudiant -->
</div>
<div class="user-box">
    <h1>
        <strong>CNE:</strong>
        <%=((Etudiant) request.getAttribute("Etudiant")).getCne()%></h1>
    <!-- Récupère et affiche le CNE de l'étudiant -->
</div>
<div class="user-box">
    <h1>
        <strong>DATE DE NAISSANCE:</strong>
        <%=((Etudiant) request.getAttribute("Etudiant")).getDNaiss()%>
    </h1>
    <!-- Récupère et affiche la date de naissance de l'étudiant -->
</div>
<div class="user-box">
    <h1>
        <strong>TELEPHONE:</strong>
        <%=((Etudiant) request.getAttribute("Etudiant")).getTele()%></h1>
    <!-- Récupère et affiche le numéro de téléphone de l'étudiant -->
</div>
</div>
</div>
</body>
</html>

```

Voila les fichiers style utilisés



Ensuite on démarre l'application dans le serveur Apache Tomcat 9 et on aura le formulaire:

quand on clique sur le bouton 'Envoyer', on aura:

PARTIE 2 : CREER UN ETUDIANT

La deuxième partie de notre application de gestion d'inscription d'étudiants en JEE consiste à ajouter un nouveau formulaire nommé "part2-form.JSP" qui permettra à l'utilisateur de saisir des informations supplémentaires sur l'étudiant, telles que la date d'inscription, la filière, le groupe, le niveau (la classe), le cycle et le nombre d'années d'inscription. Tout comme dans la première partie, nous allons suivre le modèle MVC et coder un bean pour représenter l'inscription de l'étudiant, une servlet pour récupérer les données envoyées par le formulaire et les enregistrer dans le bean, et une JSP pour afficher les données transmises par la servlet.

Bean pour représenter une inscription:

```
// Import de la classe Etudiant du package com.etudiant.beans
import com.etudiant.beans.Etudiant;

// Déclaration de la classe inscriptionBeans
public class inscriptionBeans {

    // Déclaration des variables de classe : Etudiant, date d'inscription, niveau, filière,
    groupe, cycle et nombre d'années d'inscription
    private static Etudiant etudiant;
    private static String dInscription;
    private static String niveau;
    private static String filiere;
    private static String groupe;
    private static String cycle;
    private static String nbrAnneeInscription;

    // Méthode pour récupérer l'objet Etudiant
    public static Etudiant getEtudiant() {
        return etudiant;
    }
    // Méthode pour définir l'objet Etudiant
    public static void setEtudiant(Etudiant etudiant) {
        inscriptionBeans.etudiant = etudiant;
    }
    // Méthode pour récupérer la date d'inscription
    public static String getdInscription() {
        return dInscription;
    }
    // Méthode pour définir la date d'inscription
    public static void setdInscription(String dInscription) {
        inscriptionBeans.dInscription = dInscription;
    }
    // Méthode pour récupérer le niveau
    public static String getNiveau() {
        return niveau;
    }
    // Méthode pour définir le niveau
    public static void setNiveau(String niveau) {
        inscriptionBeans.niveau = niveau;
    }
    // Méthode pour récupérer la filière
    public static String getFiliere() {
        return filiere;
    }
}
```

```
// Méthode pour définir la filière
public static void setFiliere(String filiere) {
    inscriptionBeans.filiere = filiere; }
// Méthode pour récupérer le groupe
public static String getGroupe() {
    return groupe;}
// Méthode pour définir le groupe
public static void setGroupe(String groupe) {
    inscriptionBeans.groupe = groupe;}
// Méthode pour récupérer le cycle
public static String getCycle() {
    return cycle;}
// Méthode pour définir le cycle
public static void setCycle(String cycle) {
    inscriptionBeans.cycle = cycle;}
// Méthode pour récupérer le nombre d'années d'inscription
public static String getNbrAnneeInscription() {
    return nbrAnneeInscription;}
// Méthode pour définir le nombre d'années d'inscription
public static void setNbrAnneeInscription(String nbrAnneeInscription) {
    inscriptionBeans.nbrAnneeInscription = nbrAnneeInscription;
}
}
```

L'utilisateur commencera par remplir le formulaire d'inscription en saisissant les informations requises:

```
<!doctype html>
<html lang="en">
<head>
<title>Inscription etudiant</title>

<meta charset="utf-8">
<meta name="viewport"
    content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="login-box">
        <h2>Inscription:</h2>
        <form method="post" action="part2Servlet">
            <div class="input-group">
                <div class="user-box">
                    <input type="text" class="form-control" id="firstName"
                        name="firstName" placeholder="votre prenom">
                </div>
                <div class="user-box">
```

```
<input type="text" class="form-control" id="lastName"
      name="lastName" placeholder="votre prenom">
</div>
<div class="user-box">
  <input type="email" class="form-control" id="email" name="email"
        placeholder="votre email">
</div>
<div class="user-box">
  <input type="text" class="form-control" id="cne" name="cne"
        placeholder="votre CNE">
</div>
<label class="user-box">Date de naissance:</label>
<div class="user-box">
  <input type="date" id="dNaiss" name="dNaiss">
</div>
<label class="user-box">Date de naissance:</label>
<div class="user-box">
  <input type="date" class="form-control" id="dInscription"
        name="dInscription" placeholder="votre date inscription">
</div>
<div class="user-box">
  <input type="tele" class="form-control" id="tele" name="tele"
        placeholder="votre numero de telephone">
</div>
<div class="user-box">
  <input type="number" class="form-control" id="nbrAnneeInscription"
        name="nbrAnneeInscription"
        placeholder="Nombre d'ans d'inscription">
</div>
<div class="user-box">
  <input type="text" class="form-control" id="groupe" name="groupe"
        placeholder="votre groupe">
</div>
<div class="user-box">
  <select name="niveau" class="custom-select" id="niveau">
    <option value="">Select niveau..</option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
  </select>
</div>
<div class="user-box">
  <select name="cycle" class="custom-select" id="cycle">
    <option value="">Select cycle..</option>
    <option value="preparatoire">preparatoire</option>
    <option value="d'ingenieur">d'ingenieur</option>
  </select>
</div>
```

```

        </select>
    </div>
    <div class="user-box">
        <select name="filiere" class="custom-select" id="filiere">
            <option value=" ">Select filiere..</option>
            <option value="cp">CP</option>
            <option value="iric">IRIC</option>
            <option value="iid">IID</option>
            <option value="gi">GI</option>
            <option value="ge">GE</option>
            <option value="gpee">GPEE</option>
        </select>
    </div>
    <input type="submit" value="Envoyer!">
</div>
</form>
</div>
</body>
</html>

```

Une fois les données envoyées à la servlet 'part2Servlet', elle sera chargée de les récupérer et de les stocker dans le bean qui représente l'inscription de l'étudiant:

```

package com.part2.servlets;

@WebServlet("/part2Servlet")
public class part2Servlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Retrieve the first name and last name and email from myForm.jsp
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String email = request.getParameter("email");
        String cne = request.getParameter("cne");
        String tele = request.getParameter("tele");
        String dNaiss = request.getParameter("dNaiss");
        String dInscription = request.getParameter("dInscription");
        String cycle = request.getParameter("cycle");
        String nbrAnneeInscription = request.getParameter("nbrAnneeInscription");
        String groupe = request.getParameter("groupe");
        String niveau = request.getParameter("niveau");
        String filiere = request.getParameter("filiere");

        // Create a new instance of the Etudiant and populate its properties
    }
}

```

```

Etudiant Etudiant = new Etudiant();
Etudiant.setFirstName(firstName);
Etudiant.setLastName(lastName);
Etudiant.setEmail(email);
Etudiant.setTele(tele);
Etudiant.setdNaiss(dNaiss);
Etudiant.setCne(cne);

inscriptionBeans inscription = new inscriptionBeans();
inscription.setCycle(cycle);
inscription.setdInscription(dInscription);
inscription.setEtudiant(Etudiant);
inscription.setFiliere(filiere);
inscription.setGroupe(groupe);
inscription.setNbrAnneeInscription(nbrAnneeInscription);
inscription.setNiveau(niveau);

// Store the Etudiant in the request object
request.setAttribute("inscription", inscription);

// Forward the request to the myResponse.jsp page
request.getRequestDispatcher("/part2-response.jsp").forward(request, response);
}
}

```

La servlet transmettra ensuite ces données à la 'part2-response.JSP', qui affichera la fiche d'inscription de l'étudiant créé :

```

<!DOCTYPE html>
<%@ page import="com.etudiant.beans.Etudiant"%>
<%@ page import="com.inscription.beans.inscriptionBeans"%>

<html>
<head>
<link rel="stylesheet" href="css/styleAffichage.css">
<title>my Response</title>

</head>
<body>
    <div class="login-box">
        <div class="input-group">
            <div class="user-box">
                <h1>
                    <strong>PRENOM:</strong>
                    <%=((inscriptionBeans) request.getAttribute("inscription")).getEtudiant().getFirstName()%>
                </h1>
            </div>
        </div>
    </div>

```

```

</div>

<div class="user-box">

    <h1>

        <strong>NOM:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getEtudiant().getLastName()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>EMAIL:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getEtudiant().getEmail()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>CNE:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getEtudiant().getCne()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>DATE DE NAISSANCE:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getEtudiant().getDNaiss()%>

    </div>

<div class="user-box">

    <h1>

        <strong>TELEPHONE:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getEtudiant().getTele()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>Date inscription:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getdInscription()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>Niveau:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getNiveau()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>Filiere:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getFiliere()%></h1>

    </div>

<div class="user-box">

    <h1>

        <strong>Groupe:</strong>

        <%=((inscriptionBeans) request.getAttribute("inscription")).getGroupe()%></h1>

    </div>

```

```
<div class="user-box">
  <h1>
    <strong>Cycle:</strong>
    <%=((inscriptionBeans) request.getAttribute("inscription")).getCycle()%></h1>
  </div></div></div>
</body>
</html>
```

Quand on démarre l'application dans le serveur Apache Tomcat 9 on aura:

Inscription:

ZAKARIA EL FATHI

zakelfathi@gmail.com S134073148

Date de naissance: 15/01/2000

Date d'inscription: 17/09/2019 0612539876

4 G4

2 d'ingenieur

IID

ENVOYER!

Quand on envoie, on aura:

my Response

localhost:8091/TP1%20JEE/part2Servlet

PRENOM: ZAKARIA NOM: EL FATHI

EMAIL: zakelfathi@gmail.com

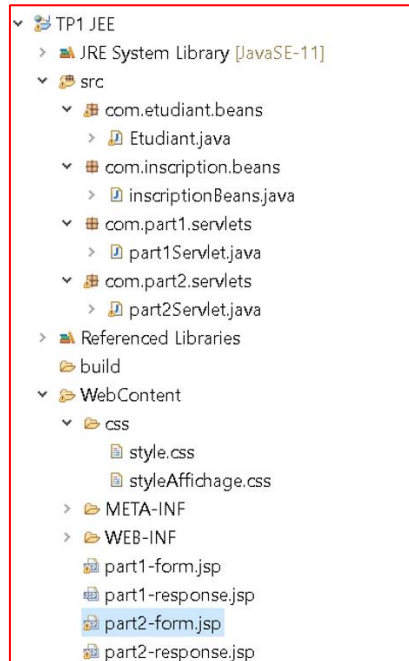
CNE: S134073148 DATE DE NAISSANCE: 2000-01-15

TELEPHONE: 0612539876 Date inscription: 2019-09-17

Niveau: 2 Filiere: iid

Groupe: G4 Cycle: d'ingenieur

dans la suite on aura l'architecture suivante pour notre projet:



CONCLUSION:

En conclusion, cet atelier nous a permis de concevoir une application de gestion d'inscription d'étudiants en JEE selon le modèle MVC en utilisant une pile technologique comprenant Eclipse, Tomcat, JSP, Servlet, JEE, HTML, et CSS. Nous avons développé un bean pour représenter un étudiant et un autre pour représenter l'inscription d'un étudiant donnée. Nous avons également implémenté une servlet pour récupérer les données envoyées par les formulaires, les enregistrer dans les beans correspondants, et les transmettre à une JSP pour afficher les fiches étudiant et d'inscription créées. Ce projet nous a permis d'approfondir notre compréhension de la mise en œuvre de l'architecture MVC en JEE et de renforcer nos compétences en développement web en utilisant diverses technologies.