

Les formulaires

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en Programmation par contrainte (IA)
Ingénieur en Génie logiciel

`elmouelhi.achref@gmail.com`

Un formulaire

- un outil graphique
 - que nous créons avec le langage de description HTML
 - que nous gérons avec un langage de programmation tel que PHP, Servlets Java...
- il permet à l'utilisateur de saisir des données
- et de les envoyer vers une autre page, vers une base de données...

JEE : formulaires

Contenu de la page AjoutPersonne.jsp

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ajout</title>
  </head>
  <body>
    <form method="post" action="AjoutPersonne">
      <div>Formulaire d'ajout d'une Personne</div>
      <div><label for="nom">Nom *</label>
        <input type="text" id="nom" name="nom" value="" />
      </div>
      <div><label for="prenom">Prénom *</label>
        <input type="text" id="prenom" name="prenom" value="" />
      </div>
      <input type="submit" value="Ajouter" />
    </form>
  </body>
</html>
```

Déclaration d'un formulaire :

```
<form method="post ou get" action="url-pattern">  
    ...  
</form>
```

Déclaration d'un formulaire :

```
<form method="post ou get" action="url-pattern">  
    ...  
</form>
```

Les attributs d'un formulaire

- **action** : indique l'url-pattern d'une servlet (tel qu'on l'a déclarée dans le `web.xml` ou avec l'annotation `@WebServlet(...)`)
- **method** : concerne l'envoi de données et peut prendre deux valeurs.
 - **get** : c'est la méthode `doGet()` de la servlet qui sera appelée.
 - **post** : c'est la méthode `doPost()` de la servlet qui sera appelée.

Par rapport à l'exemple précédent

- Quand on saisit l'url `.../AjoutPersonne` dans le navigateur, c'est la méthode `doGet()` qui sera appelée.
- Quand on valide le formulaire, il est préférable d'appeler la méthode `doPost()`.

JEE : formulaires

Les `doGet()` **et** `doPost()` **de** `PersonneServlet`

```
protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    this.getServletContext().getRequestDispatcher("/WEB-INF/AjoutPersonne
        .jsp").forward(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    String nom = request.getParameter("nom");
    String prenom = request.getParameter("prenom");
    Personne p = new Personne(nom, prenom);
    PersonneDaoImpl daop = new PersonneDaoImpl();
    daop.save(p);
    request.setAttribute("personne", p);
    this.getServletContext().getRequestDispatcher("/WEB-INF/confirmation.
        jsp").forward(request, response);
}
```

Contenu de `confirmation.jsp`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ajout</title>
  </head>
  <body>
    Person ${ prenom } ${ nom } has been
      successfully added to our database.
  </body>
</html>
```


Contrôler la validation du formulaire

- Interdire l'insertion vide (pour les deux champs nom et prénom)
- Pas de chaîne avec une longueur inférieure à deux
- Première lettre en majuscule
- Afficher un message en cas de format non-respecté
- Garder les valeurs saisies affichées dans le formulaire, en cas d'erreur

Dans la servlet, une méthode pour contrôler la saisie

```
public boolean verifChaine(String s) {  
    if (s == null || s.length() < 2)  
        return false;  
    char c = s.charAt(0);  
    if (!(c >= 'A' && c <= 'Z'))  
        return false;  
    for(int i = 0 ; i< s.length(); i++) {  
        c = s.charAt(i);  
        if (!(c >= 'a' && c <= 'z') && !(c >= 'A' && c <= 'Z'))  
            return false;  
    }  
    return true;  
}
```

Préparons le formulaire pour les nouvelles améliorations

```
<form method="post" action="AjoutPersonne">
  <div>Formulaire d'ajout d'une Personne</div>
  <div><label for="nom">Nom *</label>
    <input type="text" id="nom" name="nom" value="{
      nomSaisi }"/>
    ${ nomIncorrect }
  </div>
  <div><label for="prenom">Prénom *</label>
    <input type="prenom" id="prenom" name="prenom"
      value="{ prenomSaisi }" />
    ${ prenomIncorrect }
  </div>
  <input type="submit" value="Ajouter" />
</form>
```

Dans la servlet, une méthode pour contrôler la saisie

```
protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    String nom = request.getParameter("nom");
    String prenom = request.getParameter("prenom");
    boolean verifNom = verifChaine(nom);
    boolean verifPrenom = verifChaine(prenom);
    if (!verifNom)
        request.setAttribute("nomIncorrect", "format incorrect");
    if (!verifPrenom)
        request.setAttribute("prenomIncorrect", "format incorrect");

    if (!verifNom || !verifPrenom){
        request.setAttribute("nomSaisi", nom);
        request.setAttribute("prenomSaisi", prenom);
        this.getServletContext().getRequestDispatcher("/WEB-INF/
            AjoutPersonne.jsp").forward(request, response);
    }
    else {
        // utiliser le dao pour sauvegarder la personne
        this.getServletContext().getRequestDispatcher("/WEB-INF/
            confirmation.jsp").forward(request, response);
    }
}
```

Une deuxième solution

- Utiliser les exceptions pour afficher les erreurs

Une deuxième solution

- Utiliser les exceptions pour afficher les erreurs
- On ne modifie pas la vue

La méthode de contrôle de saisie avec utilisation d'exception

```
public void verifChaine(String s) throws Exception {
    if (s == null || s.length() < 2)
        throw new Exception("La chaîne doit comporter au
            moins deux caractères");
    char c = s.charAt(0);
    if (!(c >= 'A' && c <= 'Z'))
        throw new Exception("La chaîne doit commencer par une
            lettre en majuscule");
    for(int i = 0 ; i< s.length(); i++) {
        c = s.charAt(i);
        if (!(c >= 'a' && c <= 'z') && !(c >= 'A' && c <= 'Z'))
            throw new Exception("La chaîne ne peut contenir que
                des lettres");
    }
}
```

JEE : formulaires

```
protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    boolean test = true;
    String nom = request.getParameter("nom");
    String prenom = request.getParameter("prenom");
    try {
        verifChaine(nom);
    } catch (Exception e) {
        request.setAttribute("nomIncorrect", e.getMessage() );
        test = false;
    }
    // refaire la meme chose pour prenom
    if (test){
        // utiliser le dao pour sauvegarder la personne
        this.getServletContext().getRequestDispatcher("/WEB-INF/
            confirmation.jsp").forward(request, response);
    }
    else {
        request.setAttribute("nomSaisi", nom );
        request.setAttribute("prenomSaisi", prenom);
        this.getServletContext().getRequestDispatcher("/WEB-INF/
            AjoutPersonne.jsp").forward(request, response);
    }
}
```


Une troisième solution

- déplacer les contrôles dans une classe intermédiaire entre le bean et le contrôleur (on l'appel généralement classe métier)