

## **Formation Framework Struts1&2**

### **Atelier 4 : Création d'une application web selon le patron de conception MVC2**

Encadré par M.BOULCHAHOU

Objectif de l'atelier.....	1
Pour commencer ! .....	1
Plusieurs actions à intercepter .....	1
Mapping de plusieurs actions .....	4
Plusieurs actions dans une vue .....	5
Implémenter MVC2 .....	7
Création du contrôleur principal .....	8
Adaptation du web.xml .....	9
Adaptation de la vue accueil.jsp.....	9
Conclusion.....	10

## **Objectif de l'atelier**

*L'objectif de cet atelier est de créer une application web en se basant sur le patron de conception MVC2. Il s'agit également de comprendre la différence entre MVC et MVC2.*

*Vous allez apprendre à:*

- *Créer une application selon MVC2*
- *Différencier entre MVC et MVC2*

## **Pour commencer !**

*Pour commencer cet atelier, s'assurer que les projets « CoucheMetier » et « CouchePresentation » réalisés lors de l'atelier 3 sont opérationnel.*

## **Plusieurs actions à intercepter**

*Imaginons que nous souhaitons mettre en œuvre plusieurs actions :*

- *Rechercher un étudiant sur la base de sa carte nationale*
- *Ajouter un étudiant en saisissant un nom et une cin.*
- *Supprimer un étudiant sur la base de sa carte nationale*

## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le parton de conception MVC2

Encadré par M.BOULCHAOUB

Nous serons alors obligé de créer trois Servlet, chacune prendra en charge une action.

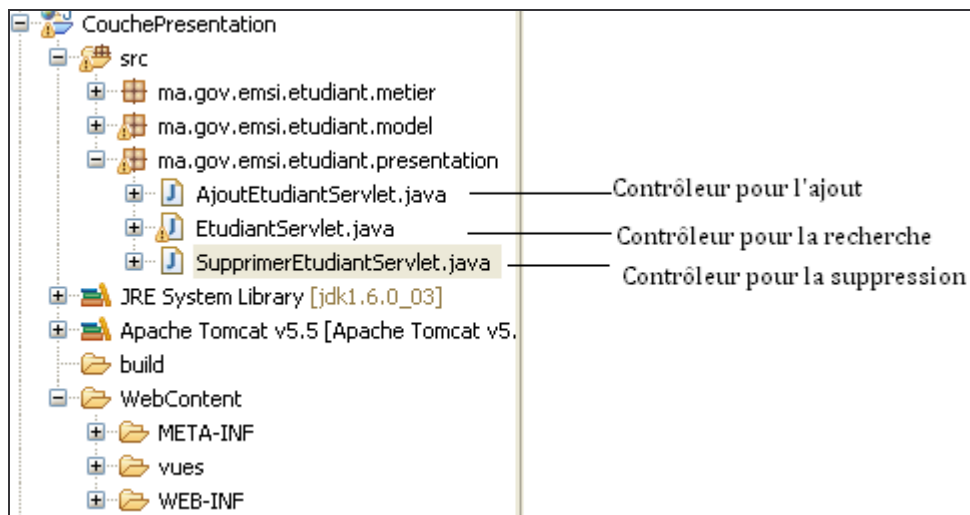
Dans le projet « CouchePresentation », vous avez déjà le contrôleur EtudiantServlet qui permet de recherche un étudiant sur la base de la CIN.

Dans le projet « CouchePresentation » et dans le package « ma.gov.emsi.etudiant.presentation » ajouter :

- Un contrôleur (Servlet) ayant le nom AjoutEtudiantServlet pour ajouter un étudiant.
- Un contrôleur (Servlet) ayant le nom SupprimerEtudiantServlet pour supprimer un étudiant.

***Donc à chaque action, on crée un contrôleur***

Vous devriez avoir :

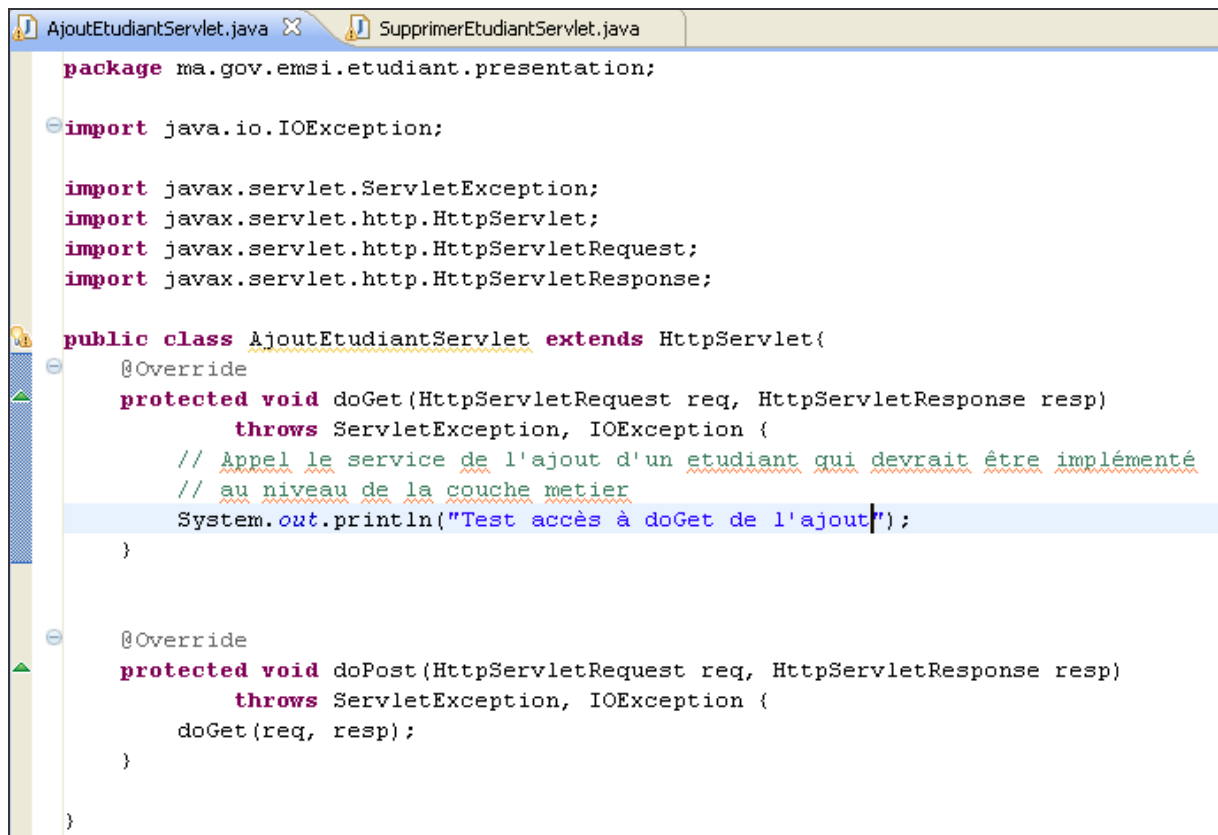


Pour le contrôleur de l'ajout :

## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le patron de conception MVC2

Encadré par M.BOULCHAHOUB



```
package ma.gov.emsi.etudiant.presentation;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class AjoutEtudiantServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        // Appel le service de l'ajout d'un etudiant qui devrait être implémenté
        // au niveau de la couche metier
        System.out.println("Test accès à doGet de l'ajout");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        doGet(req, resp);
    }
}
```

*Pour le contrôleur de la suppression :*

## **Formation Framework Struts1&2**

### *Atelier 4 : Création d'une application web selon le patron de conception MVC2*

*Encadré par M.BOULCHAHOUB*

```
package ma.gov.emsi.etudiant.presentation;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class SupprimerEtudiantServlet extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        // Appel le service de la suppression qui devrait être implémenté
        // au niveau de la couche métier
        System.out.println("Test accès à doGet de la suppression");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        doGet(req, resp);
    }
}
```

## **Mapping de plusieurs actions**

*Au niveau du fichier web.xml, faites le mapping entre les trois contrôleurs ci-dessus et leurs actions :*

*Vous aurez besoin de trois balises <servlet> et trois balises <servlet-mapping>.*

## **Formation Framework Struts1&2**

### **Atelier 4 : Création d'une application web selon le patron de conception MVC2**

**Encadré par M.BOULCHAOUB**

```
<servlet>
  <servlet-name>RechercheEtudiant</servlet-name>
  <servlet-class>ma.gov.emsi.etudiant.presentation.EtudiantServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>AjoutEtudiant</servlet-name>
  <servlet-class>ma.gov.emsi.etudiant.presentation.AjoutEtudiantServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>SupprimerEtudiant</servlet-name>
  <servlet-class>ma.gov.emsi.etudiant.presentation.SupprimerEtudiantServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>RechercheEtudiant</servlet-name>
  <url-pattern>/rechercher</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>AjoutEtudiant</servlet-name>
  <url-pattern>/ajouter</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>SupprimerEtudiant</servlet-name>
  <url-pattern>/supprimer</url-pattern>
</servlet-mapping>
```

## **Plusieurs actions dans une vue**

*Dans la page accueil.jsp nous allons créer trois fonctions javascript pour déclencher trois actions (l'ajout, la suppression et la recherche).*

## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le patron de conception MVC2

Encadré par M.BOULCHAHOUB

```
<script type="text/javascript">
function fctAjout() {
    this.document.forms[0].action="./ajouter";
    this.document.forms[0].submit();
}

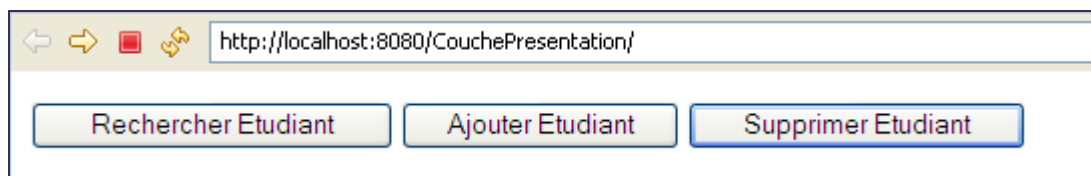
function fctRecherche() {
    this.document.forms[0].action="./rechercher";
    this.document.forms[0].submit();
}

function fctSupprimer() {
    this.document.forms[0].action="./supprimer";
    this.document.forms[0].submit();
}
</script>
<body>

<form method="get">
<input type="submit" value="Rechercher Etudiant" onclick="fctRecherche()" />
<input type="submit" value="Ajouter Etudiant" onclick="fctAjout()" />
<input type="submit" value="Supprimer Etudiant" onclick="fctSupprimer()" />
</form>
```

Déployer votre projet sur le serveur Tomcat.

Vous devriez avoir trois boutons, chacun correspond à une action :



### Problématique :

**Pour les traitements communs aux actions (traces, sécurité...) nous sommes obligés de parcourir tous les contrôleurs de notre projets pour implémenter le traitement communs.**

Exemple simple :

Nous souhaitons afficher la phrase « Action déclenchée » une fois une action est déclenché par un utilisateur. Que proposer vous ?

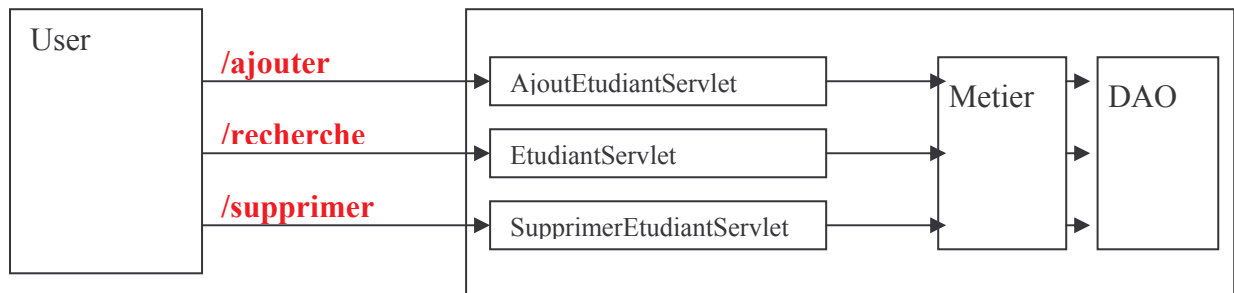
## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le patron de conception MVC2

Encadré par M.BOULCHAOUB

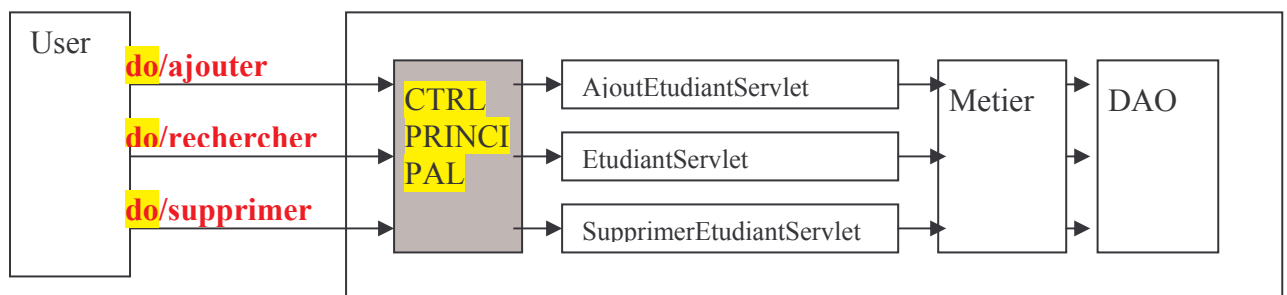
## Implémenter MVC2

*L'architecture actuelle de notre projet est la suivante :*



*On dit qu'une application basé sur MVC comporte plusieurs points d'entrées.*

*On devrait changer l'architecture de notre projet comme suivant :*



*Il s'agit d'introduire un nouveau composant, c'est **le contrôleur principal** qui aura comme missions :*

- Intercepter toutes les actions
- Dispatcher vers les sous contrôleurs
- Implémenter les traitements communs aux actions

*Pour passer du MVC à MVC2, vous devriez :*

- Créer le contrôleur principal dans votre projet
- Adapter le web.xml pour permettre au contrôleur d'intercepter toutes les actions.
- Adapter la vue accueil.jsp

## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le parton de conception MVC2

Encadré par M.BOULCHAHOU

## Création du contrôleur principal

Le contrôleur principal sera une servlet qui intercepte la requête et renvoie vers le sous contrôleur qui traite l'action demandé. Supposons que le nom de ce contrôleur est ServletAction.

```
package ma.gov.emsi.etudiant.presentation;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletAction extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp)

        throws ServletException, IOException {
        // on récupère la méthode d'envoi de la requête
        String methode = req.getMethod().toLowerCase();
        // on récupère l'action à exécuter
        String action = req.getPathInfo();

        // exécution action
        if (methode.equals("get") && action.equals("/rechercher")) {
            // recherche d'un etudiant
            EtudiantServlet sousCtrlSearch = new EtudiantServlet();
            sousCtrlSearch.doGet(req, resp);
            return;
        }
        if (methode.equals("get") && action.equals("/ajouter")) {
            // ajout d'un etudiant
            AjoutEtudiantServlet sousCtrlAjout = new
AjoutEtudiantServlet();
            sousCtrlAjout.doGet(req, resp);
            return;
        }
        if (methode.equals("get") && action.equals("/supprimer")) {
            // suppression d'un etudiant
            SupprimerEtudiantServlet sousCtrlSuppression = new
SupprimerEtudiantServlet();
            sousCtrlSuppression.doGet(req, resp);
            return;
        }

    }

    @Override
```



## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le patron de conception MVC2

Encadré par M.BOULCHAHOUB

```
protected void doPost(HttpServletRequest req, HttpServletResponse
resp)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(req, resp);
}
}
```

## Adaptation du web.xml

Même si on a plusieurs actions dans le web.xml nous aurons **uniquement** le mapping relatif au contrôleur principal. L'expression régulière /do/\* permet au contrôleur principal d'intercepter toutes les actions qui commencent par /do

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>GestionEtudiant</display-name>

    <servlet>
        <servlet-name>CtrlPrincipal</servlet-name>
        <servlet-
class>ma.gov.emsi.etudiant.presentation.ServletAction</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>CtrlPrincipal</servlet-name>
        <url-pattern>/do/*</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>vues/accueil.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

## Adaptation de la vue accueil.jsp

Modifier les fonctions javascript en faisant précéder les actions par /do/

## Formation Framework Struts1&2

### Atelier 4 : Création d'une application web selon le patron de conception MVC2

Encadré par M.BOULCHAOUB

```
<script type="text/javascript">
function fctAjout() {
    this.document.forms[0].action="./do/ajouter";
    this.document.forms[0].submit();
}

function fctRecherche() {
    this.document.forms[0].action="./do/rechercher";
    this.document.forms[0].submit();
}

function fctSupprimer() {
    this.document.forms[0].action="./do/supprimer";
    this.document.forms[0].submit();
}
</script>
```

## Conclusion

MVC2 est un patron de conception qui permet à l'application d'avoir un seul point d'entrée, c'est le contrôleur principal qui a comme mission l'interception de toutes les actions qui répondent à l'expression régulière définie au niveau du web.xml

## Fin de l'atelier