TP 4 : Struts :

Exercice 1 :

2°)

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class Hello

\*/

**public** **class** Hello **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** Hello() {

**super**();

// **TODO** Auto-generated constructor stub

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

response.getWriter().append("Hello world ").append(request.getContextPath());

}

/\*\*

\* **@see** HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

doGet(request, response);

}

}

3°) l’annotation présente : @WebServlet("/Hello")

4°) Il est nécessaire d’ajouter :

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>fr.esigelec.Hello</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>\*.php</url-pattern>

</servlet-mapping>

Résultat :



5°) Il est nécessaire d’ajouter :

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>fr.esigelec.Hello</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

Résultat :



6°)



7°)Cet exercice nous permet de voir l’utilisation de mapping des servlets.

On redirige tous les urls terminant par .do, vers un unique servlet ici Hello.

Il s’agit de « l’initiation » pour comprendre le principe de struts.

Exercice 2 :

4°)

* Controller :
  + Le package Controller contient tous les « Controller » du site qui vont permettre d’exécuter les fichiers. Toute les classes qui seront présentes dans ce package hériterons de la classe Action de Struts.
* Dao :
  + Il s’agit du package qui va contenir les interfaces qui vont définir les différentes méthodes pour le classe DAO en fonction du type de connexion (Hibernate, JDBC)
* Dao.JDBC :
  + Contient toutes les implémentations et toutes les différentes classes nécessaires au fonctionnement du projet via l’utilisation de JDBC.
* DAO.Hibernate :
  + Contient toutes les implémentations des précédentes interfaces définit dans le package DAO, ainsi que toutes les classes nécessaires au fonctionnement du projet via l’utilisation d’Hibernate.
* DTO :
  + Contient les classes métiers correspondant aux interfaces.
* Forms :
  + Contient toutes les classes qui vont être reliées au form dans le fichier JSP. Les différents liens seront faits dans le fichier de configuration.
* Business :
  + Ce package est pour le moment vide.

Les fichiers suivants servent :

* Web.xml :
  + Permet de définir les différents paramètres pour l’utilisation du site. Il va également permettre la déclaration des différents servelts, form, ainsi que les patterns d’URL.
* Struts-config.xml
  + Contient les différents cas et type de retour possible lors de la validation ou non d’un formulaire, ainsi que la déclaration des différentes classes pour les formes.
* Bdd.properties :
  + Définit les différents éléments pour permettre la connexion à la base de données (URL de connexion, Login, Driver nécessaire, password, … etc).
* mesMessages.properties :
  + Permet de définir les messages d’erreur ou le texte qui s’affichera dans la page en permettant d’avoir une personnalisation en fonction de la langue par exemple.

5°)

A°) Une Connexion Pool est une classe permettant la gestion de la connexion vers la base de données de plusieurs utilisateur en même temps en utilisant un pool de connexion.

B°) Un pool de connexion va générer et gérer un certain nombre de connexion ouverte et prête à l’emploi par les futur utilisateurs. Il ne sera donc pas nécessaire de faire toute l’étape d’ouverture et de connexion pour l’utilisateur. La Connexion sera déjà prête.

C°) La méthode close de la connexion va fermer la connexion courante de l’utilisateur / contexte.