

TDM04 de Technologie Web 2: Framework J2EE (Struts 2)

ASI4 - INSA Rouen
CORRECTION

WikiStruts

Le but de cet exercice est de mettre en place un dictionnaire en ligne, utilisant Struts 2.

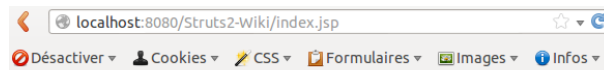
A partir d'une application vide, ou de n'importe quel autre exemple fourni en cours, vous construirez les grandes lignes d'un dictionnaire en ligne, style Wikipédia, permettant de faire une requête sur un mot afin d'en afficher la définition.

Vous développerez aussi bien le code nécessaire à la gestion des requêtes de définition que celui concernant la saisie de nouveaux mots et de nouvelles définitions.

La page principale de votre application (`index.jsp`) permettra à l'utilisateur de saisir le terme à rechercher. Une vérification du champs sera effectuée côté serveur (champs non vide requis).

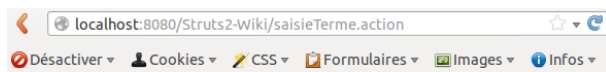
Vous pourrez utiliser la BD fournie en annexe si vous le souhaitez.

Pensez à bien développer votre application en MVC !



Saisir le terme a rechercher

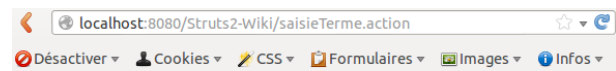
Terme :



Saisir le terme a rechercher

Veillez saisir un Terme.

Terme :



Saisir le terme a rechercher

Terme :

ASI: Chouette département de l'INSA

Remarques

1. Les applications J2EE nécessitent un serveur d'application web J2EE (*ex* : Tomcat ou JBoss/WildFly). Un tel serveur doit être installé **en local** sur votre machine en salle de TP, dans le répertoire **/tmp**. Au cours de ce TP vous utiliserez WildFly dont vous pourrez récupérer l'archive dans le répertoire **/uv/asi/technoweb/**. Vérifiez bien que vos variables systèmes **\$JBOSS_HOME** et **\$JAVA_HOME** pointent vers les bons répertoires.
2. Pensez à vérifier vos pages HTML et CSS sur le site du W3C (<http://validator.w3.org/>) ou n'importe quel autre validateur.
3. À l'issu de la séance, vous aurez accès à la correction de ce TDM au format PDF.

Correction

Déploiement

```
Struts2-Wiki.war
├─ index.jsp
├─ META-INF
├─ WEB-INF
│   └─ web.xml
│       └─ lib
│           └─ struts2-core-2.2.1.jar, ...
└─ classes
    └─ struts.xml, ...
        └─ wikistruts
            └─ model
                └─ Entite.class, BD.class
            └─ action
                └─ SaisirTerme.class
```

index.jsp

```
<?xml version="1.0" encoding="UTF-8" ?>
<%@ taglib prefix="s" uri="/struts-tags" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Recherche d'un terme</title>
    <s:head />
  </head>
  <body bgcolor="#ffffff">
    <h1>Saisir le terme à rechercher</h1>
    <s:form action="saisirTerme">
      <s:textfield name="termeBean.terme" label="Terme" />
      <s:submit />
    </s:form>
    <p><s:property value="termeBean" /></p>
  </body>
</html>
```

ajouter.jsp

```
<?xml version="1.0" encoding="UTF-8" ?>
<%@ taglib prefix="s" uri="/struts-tags" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Ajout d'un terme et de sa définition</title>
    <s:head />
  </head>
  <body bgcolor="#ffffff">
    <h1>Saisir le terme et sa définition à ajouter</h1>
    <s:form action="ajoutTerme">
      <s:textfield name="termeBean.terme" label="Terme" />
      <s:textfield name="termeBean.definition" label="Définition" />
      <s:submit />
    </s:form>
    <p><s:property value="termeBean" /></p>
  </body>
</html>
```

struts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
" http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <constant name="struts.devMode" value="true" />
  <package name="basicstruts2" extends="struts-default">
    <!-- If no class attribute is specified the framework will assume success and render
the result index.jsp -->
    <!-- If no name value for the result node is specified the success value is the
default -->
    <action name="index">
      <result>/index.jsp</result>
    </action>
    <action name="saisirTerme" class="wikistruts.action.SaisirTerme" method="execute">
      <result name="input">/index.jsp</result>
      <result name="success">/index.jsp</result>
    </action>
  </package>
</struts>
```

```

</action>
<action name="ajoutTerme" class="wikistruts.action.SaisirTerme" method="execute">
  <result name="input">/ajouter.jsp</result>
  <result name="success">/ajouter.jsp</result>
</action>
</package>
</struts>

```

SaisirTerme.java

```

package wikistruts.action;

import com.opensymphony.xwork2.ActionSupport;
import wikistruts.model.Entite;
import wikistruts.model.BD;

public class SaisirTerme extends ActionSupport {
    private static final long serialVersionUID = 1L;
    private Entite termeBean;

    public Entite getTermeBean() {
        return this.termeBean;
    }

    public void setTermeBean(Entite entite) {
        this.termeBean = entite;
    }

    public String execute() throws Exception {
        if(this.termeBean!=null) {
            if(this.termeBean.getDefinition().equals(""))
                this.termeBean.setDefinition(BD.getDefinition(this.termeBean.getTerme()));
            else
                BD.setDefinition(this.termeBean.getTerme(), this.termeBean.getDefinition());
            return SUCCESS;
        } else
            return INPUT;
    }

    public void validate(){
        if(this.termeBean!=null && this.termeBean.getTerme().length() == 0 )
            addFieldError("termeBean.terme", "Veuillez saisir un Terme.");
    }
}

```

BD.java

```

package wikistruts.model;

import java.util.HashMap;
import java.util.Iterator;

public class BD {
    private static HashMap<String,String> listeDeTermes = new HashMap<String,String>();

    public static void setDefinition(String terme, String definition) {
        if(terme!=null && definition != null && !terme.equals("") && !definition.equals(""))
            BD.listeDeTermes.put(terme, definition);
    }

    public static String getDefinition(String terme) {
        String definition = "Terme non trouvé";
        if(BD.listeDeTermes.containsKey(terme))
            definition = BD.listeDeTermes.get(terme);
        return definition;
    }
}

```

Entite.java

```

package wikistruts.model;

public class Entite {
    private String terme = "";
    private String definition = "";

    public String getTerme() {
        return this.terme;
    }

    public String getDefinition() {
        return this.definition;
    }

    public void setTerme(String terme) {
        this.terme = terme;
    }
}

```

```
public void setDefinition(String definition) {
    this.definition = definition;
}
public String toString() {
    if(this.getTerme().length()>0 && this.getDefinition().length()>0)
        return this.getTerme() + ": " + this.getDefinition();
    else
        return "";
}
}
```