# 概述

Struts2 是基于MVC的WEB框架。

# Demo

2. Hello struts

(1) eclipse中创建Web动态项目struts

(2) 导入jar包(WEB-INF/lib)

(3) 新建web.xml

在WEB-INF目录下新建web.xml,其中配置了一个 Filter, 所有的请求都被过滤给了这个 Filter org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter

<web-app>

<filter>

<filter-name>struts2</filter-name>

<filter-class>

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>struts2</filter-name>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

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

</filter-mapping>

</web-app>

(4) 配置struts.xml

在src目录下创建一个struts.xml文件:

<?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>

<package name="basicstruts" extends="struts-default">

<action name="index">

<result>index.jsp</result>

</action>

</package>

</struts>

(5) 创建index.jsp

在web目录下创建index.jsp:

hello Struts2

(6) 部署在tomcat中，重启测试

部署在Tomcat中，重启tomcat，然后访问地址，观察效果

http://127.0.0.1:8080/struts/index

2. 基本思路

(1). 所有的访问都会被web.xml中配置的Struts 的 Filter所拦截

(2). 拦截之后，就进入struts的工作流程

(3). 访问的地址是/index,根据struts按照 struts.xml中的配置，服务端跳转到index.jsp

(4). 显示index.jsp的内容

3. 显示数据到jsp

(1). Product.java 用于存放数据

Model层使用一个简单的Product用于存放数据:

package com.how2java.bean;

public class Product {

int id;

String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

(2). ProductAction用于控制

package com.how2java.action;

import com.how2java.bean.Product;

public class ProductAction {

private Product product;

public String show() {

product = new Product();

product.setName("iphone7");

return "show";

}

public Product getProduct() {

return product;

}

public void setProduct(Product product) {

this.product = product;

}

}

(3). 在struts.xml中配置跳转

该配置表示 访问路径/showProduct会调用 ProductAction 类的 show 方法服务端跳转到show.jsp

<?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>

<package name="basicstruts" extends="struts-default">

<action name="showProduct" class="com.how2java.action.ProductAction" method="show">

<result name="show">show.jsp</result>

</action>

</package>

</struts>

(4). show.jsp

在web目录下创建show.jsp文件

通过EL表达式，取出product的name

注： ${product}会访问对应的Action的 getProduct()方法

注： 严格地说，是Struts通过getProudct()方法返回product对象，然后再把product对象通过request.setAttribute("product", product) 放在"product" 这个key上，这样就方便EL表达式去取出来了

<%@page isELIgnored="false"%>

${product.name}

原理：

1. 访问路径 /showProduct

2. 所有访问都被struts的filter所拦截，并进入struts的工作流程

3. 根据配置文件struts.xml, 会执行ProductAction的show方法

4. 在show方法中，将实例属性product指向一个新的对象，并设置名称为iphone7

5. 服务端跳转到show.jsp

6. 在show.jsp中，访问ProductAction.getProduct() 获取实例属性product,并显示其名称iphone7

4. 提交数据到ACTION

（1）addProduct.jsp

<html>

<form action="addProduct">

<input type="text" name="product.name">

<br/>

<input type="submit" value="submit">

</form>

</html>

（2）配置struts.xml

为addProduct路径配置对应的action和方法以及返回：

<?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>

<package name="basicstruts" extends="struts-default">

<action name="showProduct" class="com.how2java.action.ProductAction" method="show">

<result name="show">show.jsp</result>

</action>

<action name="addProduct" class="com.how2java.action.ProductAction" method="add">

<result name="show">show.jsp</result>

</action>

</package>

</struts>

（3）productAction增加add方法

在addProduct.jsp中提交数据的field是product.name

会自动调用对应的Action的setProduct(Product product)方法进行数据的注入

所以ProductAction必须提供setProduct(Product product)方法

在执行到第14行的时候，jsp传过来的name已经被注入进了属性product中了

package com.how2java.action;

import com.how2java.bean.Product;

public class ProductAction {

private Product product;

public String show(){

product = new Product();

product.setName("iphone7");

return "show";

}

public String add(){

System.out.println("product.name:"+product.getName());

return "show";

}

public Product getProduct() {

return product;

}

public void setProduct(Product product) {

this.product = product;

}

}

原理：

1. 访问路径 /addProduct

2. 所有访问都被struts的filter所拦截，并进入struts的工作流程

3. 根据配置文件struts.xml, 会执行ProductAction的add方法

4. 在add方法执行之前，Struts生成一个新的product对象，并把页面传递过来的name设置在该对象上，接着把该对象通过setProduct()方法，注入ProductAction

5. 服务端跳转到show.jsp

6. 在show.jsp中，访问ProductAction.getProduct() 获取注入的product,并显示其名称