有SetAttribute()和getAttribute()方法的对象？

HttpServletRequest/HttpSession/ServletContext

ServletContext:是一个全局对象。该对象中也有SetAttribute()和getAttribute()方法

ServletContext对象在tomcat启动时为部署在tomcat上每个web工程都创建一个ServletContext对象

ServletContext对象什么被销毁？ 当服务关闭或重启时会销毁ServletContext对象

ServletContext:是一个全局对象,可以在本工程中共享资源。

在一个Servlet中给ServletContext设置了键值对，在本工程中的任何一个Servlet中都可以通过ServletContext去获取数据

在Servlet中我们学了几个对象？

ServletConfig 读取初始化资源

HttpServletRequest 请求对象

HttpServletResponse 响应对象

HttpSession 会话对象

ServletContext 全局对象

HttpServletRequest/HttpSession/ServletContext这三个对象按生命周期的大小排序？

HttpServletRequest-->HttpSession-->ServletContext

Jsp（Java Server Pages）技术,为了减化Servlet。

JSP是一个特殊的Servlet.

一个jsp文件必须有一个page指令

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

Jsp文件包含两部分内容:

(1)html标签的静态内容

(2)jsp脚本

脚本必须放到<%

//Java代码

%>中

输出脚本中定义变量的值可以使用<%=变量名%>

<%!

//Java代码,都会被当成全成变量使用

//只能写变量的定义，只能定义变量或方法

int a=100;

int b=0;

%>

Jsp内置对象 out request response session

----------------------------回顾----------------------------------------------

JSP原始代码中包含了JSP元素和Template(模板,Html) data两类 ：

Template data指的是JSP引擎不处理的部分，即标记<%……%>以外的部分，例如代码中的HTML的内容等 ，这些数据会直接传送到客户端的浏览器。

JSP元素则是指将由JSP引擎直接处理的部分，这一部分必须符合JSP语法，否则会导致编译错误。

<%! %>用于定义变量和方法！在此定义的变量和方法属于全局的变量和方法。

<%!

int age=22;

String name=”lisi”;

//错误age=age+2;

%>

<% %> 包含变量的声明，表达式，和程序逻辑。此变量是SERVLET中的方法变量。

<%

String str= "张三";

Int age = 20;

boolean isRead= false;

//写其它Java代码

>%

<%=表达式%>表示输出

<%=age%>

<%="hello".toUpperCase()%>

Jsp九中内置对象:

out Servlet中的response.getWriter()

request Servlet中的HttpServletRequest

response Servlet中的HttpServletResponse

session Servlet中的HttpSession request.getSession()

application相当于Servlet中ServletContext

config Servlet中的ServletConfig

page 代表当前页面

pageContext 页面的上下文对象,在jsp中是一个管家对象

exception 异常对象

pageContext可以获取所有的对象

<%

//输出流对象

JspWriter jw=pageContext.getOut();

jw.println("ok");

//可以获取请对象

HttpServletRequest req=(HttpServletRequest)pageContext.getRequest();

//可以获取响应对象

HttpServletResponse resp=(HttpServletResponse)pageContext.getResponse();

//获取当前页面对象

pageContext.getPage();

//全局对象application

ServletContext sc=pageContext.getServletContext();

%>

如果页面中使用exception对象，jsp中的page指令的isErrorPage="true"

JSP脚本注释

//content 单行注释

/\* content \*/ 注释一行或多行

<%-- content --%> 可注释HTML等。

jsp指令:

page指令

<%@ page page的属性设置%>

<%@ page language="java" %>

language 语言，表示jsp页面命名用的脚本语言

contentType="text/html; charset=UTF-8" 表示jsp页面内容及编码

pageEncoding="UTF-8" Jsp页面和脚本中所使用的编码

buffer 指定out对象使用缓冲区的大小

autoFlush="true" 当缓存区满了之后，就输出缓冲区中数据

errorPage="true" 代表是一个错误页面，可以直接使用exception对象

isErrorPage="xxx.jsp" 代表这个页面如果出现异常就内部转以到xxx.jsp

import 导入它其包中的类,如果导入多个用逗号来隔开

session="true" 代表在jsp脚本中可以直接使用session对象.默认值就是true

isThreadSafe="true" 代表是线程安全的

isELIgnored="false" 能够解析el表达式

include指令:

把一个页面中的源代码包含到加一个页面中

<h1>结果页面</h1>

<%

String name="zhangsan";

%>

<%=name%>

<table border="1">

<tr>

<td>当前日期:</td>

<td>

<%@ include file="include01.jsp"%>

</td>

</tr>

</table>

相当于把include01.jsp中的源码包启到include02.jsp中指定位置，一块生成Servlet

结果包含，使用是jsp标签 <jsp:include page="include03.jsp"></jsp:include>

表示include03.jsp运行后的结果被包含到了include04.jsp的指定位置

taglib指令: 引入标签库指令，稍后再讲解。

page-->request-->session-->application

<%

pageContext.setAttribute("age", 22);

request.setAttribute("age", 20);

session.setAttribute("age", 28);

session.setAttribute("hobby", "游泳");

application.setAttribute("age", 30);

application.setAttribute("address", "天津");

%>

pageContext:<%=pageContext.getAttribute("age")%><br>

request:<%=request.getAttribute("age")%><br>

session:<%=session.getAttribute("age")%><br>

application:<%=application.getAttribute("address")%><br>

1表示pageContext范围<br>

2表示request范围<br>

3表示session范围<br>

4表示application范围<br>

<%=pageContext.getAttribute("age",4) %>

----------------------------------------------------------------

El表达式和jstl标签库

El表达式是JavaEE提供从四种作用域获取数据的功能。简化我们取值操作。

el表达式的语法:${key}

由低范围到高范围逐一搜索，只要找到就显示，如果四个范围都没有则值(空字符串)""

<%

pageContext.setAttribute("age", 22);

request.setAttribute("age", 20);

request.setAttribute("name", "张三");

session.setAttribute("age", 28);

session.setAttribute("hobby", "游泳");

application.setAttribute("age", 30);

application.setAttribute("address", "天津");

%>

page-->request-->session-->application<br>

可以使用el表达式取值。<br>

<%=pageContext.getAttribute("age") %><br>

${age} //默认情况下从Page范围取key为age的值<br>

<%=pageContext.getAttribute("name") %><br>

${name} <br>

${address}

//从指定范围取某个key的值<br>

${pageScope.age}<br>

${requestScope.age} <br>

${sessionScope.age} <br>

${applicationScope.age} <br>

如果你从指定范围去取某个key，如果该key在该范围不存就显示"",不在向高范围去取值

${requestScope.name} <br>

${sessionScope.name} <br>

jstl是一个标签库,提供了很多功能，例如:循环,判断、把一个值放到指定的范围,格式化数据

（1）导入jstl相关的jar包

jstl.jar

standard.jar

(2)在jsp页面中导入标签库

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<c:set var="age" value="22"></c:set>等价于

<%

pageContext.setAttribute("age", 22);

%>