JSP概述

2019年5月22日 星期三 9:01

1. JSP概述

由于在servlet中书写前台页面代码非常复杂，于是Sun提出一门新的技术--jsp，将所有的页面内容重新书写到一个文件中，在这个文件中还可以书写java语句，这样的一个文件就称之为一个jsp文件。

在jsp页面内开发HTML代码十分便利，还可以在其中嵌入java语句，使开发变得非常简洁，直接在页面中就完成了页面构建和代码逻辑的全部内容。

2. JSP原理

在jsp页面被初次访问的时候，会被jsp翻译引擎翻译成一个servlet文件，这个文件是一个普通java文件，继承了一个具有servlet接口的类，所以当前文件才会成为一个servlet文件。

实验：寻找如下文件：

在[tomcat]/work/localhost/day14/index_jsp.java文件

JSP语法

2019年5月22日 星期三 9:33

1. JSP语法
   a. 模板元素

jsp页面中的HTML内容会在浏览器发生访问的时候，被jsp翻译引擎翻译为Servlet文件的模板元素。

```
<body>
  <div>aaaa</div>
</body>
```

```
out.write("   <body>\r\n");
out.write("     <div>aaaa</div>\r\n");
out.write("   </body>\r\n");
```

其实就是将HTML内容使用out.write()输出.

   b. 脚本表达式

语法：<%= 脚本表达式(具有返回值的表达式，或一个直接量) %>
在jsp页面被访问的时候，jsp翻译引擎将脚本表达式中的内容翻译为了原封不动的表达式结果值。

```
<h3>脚本表达式</h3>
<%=request.getContextPath() %>
</body>
```

```
out.write("     ");
out.print(request.getContextPath() );
out.write("\r\n");
```

翻译后的结果值直接放在servlet文件的对应位置，并使用out.print()输出表达式的值。

   c. 脚本片段

语法：<% 脚本片段(java语句) %>
在JSP页面被访问的时候，其中脚本片段中的内容会原封不动的放在翻译后的servlet文件制定位置。

```
<h3>脚本表达式</h3>
<%=request.getContextPath() %>
<h3>脚本片段</h3>
<%
    for(int i=0;i<5;i++){
    %>

    <font color='red'>I am li shuai;</font>

    <%
    }

 %>
```

```
    for(int i=0;i<5;i++){

  out.write("\r\n");
  out.write("   \t\t\r\n");
  out.write("   \t\t<font color='red'>I am li shuai;</font>\r\
n");
  out.write("   \t\t\r\n");
  out.write("   \t\t");

    }
```

在jsp页面中脚本片段可以分开书写，分开书写脚不片段必须要保证java语法完整，因为在jsp页面中的java语句会被翻译到servlet文件中，需要保证java语句在servlet文件中的完整性。

d. JSP声明

语法：<%! JSP声明 %>

在jsp页面被访问的时候，其中的jsp声明会被jsp翻译引擎翻译为当前servlet的成员变量，成员方法，静态代码块等内容。

```
<h3>JSP声明</h3>
<%! int i =0; %>
<%! public void m(){} %>
<%!static{} %>
```

```
public final class index_jsp
HttpJspBase
    implements org.apache.ja

 int i =0;
 public void m(){}
static{}
```

被翻译后的成员内容都会出现在类内方法外，作为成员内容使用。

e. JSP注释

<%-- JSP注释--%>

<%//abc%>	被翻译成java注释内容，由于是注释内容所以不会输出任何内容
<!-- def-->	HTML注释被翻译成模板元素，由于是HTML注释，所以不会输出任何内容，但是

	仍然会出现在页面中。
<%--ghi--%>	在翻译过程中直接被抛弃，不会出现在翻译后的servlet文件中。

f. JSP指令

语法：<%@指令名称 %>

指令本身不会产生任何的输出内容，是用来控制jsp翻译引擎如何来翻译jsp页面的。

指令内容：

page	<%@page %>　用户指定当前页面依赖哪些配置翻译页面。
include	<%@include　%>
taglib	<%@taglib　%>

page指令：

language="java"	当前JSP使用的开发语言
extends="package.class"	当前jsp翻译成servlet后要继承的类，注意此值必须是一个servlet的子类，一般情况下不要改
import="{package.class | package.*}, ..."	导入需要使用到的包 java.lang.*;javax.servlet.*;javax.servlet.jsp.*;javax.servlet.http.*;
session="true | false"	用来指定当前页面是否使用session，如果设置为true，则翻译过来的servlet中将会有对session对象的引用，于是可以直接在jsp中使用session隐式对象。但是这将导致一旦访问jsp就会调用request.getSession()方法，可能导致不必要的空间浪费。如果确定jsp中不需要session可以设为false
[buffer="none | 8kb | sizekb"]	out隐式对象所使用的缓冲区的大小
[autoFlush="true | false"]	out隐式对象是否自动刷新缓冲区，默认为true，不需要更改
[isThreadSafe="true | false"]	翻译过来的servlet是否实现SingleThreadModel
[errorPage="relative_url"]	如果页面出错，将要跳转到的页面，除了在jsp中使用此属性指定错误页面外也可以在web.xml中配置整个web应用的错误页面，如果两个都设置则jsp中的此属性起作用 <error-page> <error-code>404</error-code> <location>/error/404.jsp</location> </error-page> <error-page> <error-code>500</error-code> <location>/error/500.jsp</location> </error-page>
[isErrorPage="true | false"]	如果设置此属性为true,翻译过来的servlet中将含有Exception隐式对象,其中封装的就是上一个页面中抛出的异常对象
[contentType="mimeType [;charset=characterSet]" | "text/html ; charset=ISO-8859-1"]	和jsp乱码相关的指令,用来指定jsp输出时,设置的Content-Type响应头用来指定浏览器打开的编码
[pageEncoding="characterSet | ISO-8859-1"]	服务器翻译jsp时使用的编码集.如果向防止jsp乱码,应该保证文件的保存编码和jsp翻译成servlet用的编码以及输出到浏览器后浏览器打开的编码一致.此属性一旦设置好,翻译引擎会间接帮我们设置content-type属性.

g. include指令

在当前的jsp页面中引入其他的jsp页面，将组合之后的页面输出在浏览器。
<%@include file=""%>

h. taglib指令

标签技术
<%@taglib %>

2. 九大隐式对象

在jsp翻译成一个servlet后，其中就会包含9大隐式对象，其中Exception、session需要手动开启。在每一个jsp页面中都会有自己独立的9大隐式对象，每一个隐式对象所具有的变量名可以直接使用，在自行定义新的变量时 应避免与已有隐式对象名称重复。

面试题经常考：

九大隐式对象：

page --- 代表当前Servlet的对象
request --- 代表当前请求的对象
response --- 代表当前响应的对象
session --- 代表当前会话的对象
application --- 代表当前web应用的对象
config --- 代表当前servlet配置信息的对象
exception --- 代表当前页面产生的异常信息对象
out --- 代表当前jsp页面向浏览器输出数据的对象
pageContext --- 代表当前jsp页面上下文的对象

3. PageContext详解

a. 功能一：可以作为其他八大隐式对象的入口。

getException方法返回exception隐式对象
getPage方法返回page隐式对象
getRequest方法返回request隐式对象
getResponse方法返回response隐式对象
getServletConfig方法返回config隐式对象
getServletContext方法返回application隐式对象
getSession方法返回session隐式对象
getOut方法返回out隐式对象

b. 功能二：域对象

相关的方法：

setAttribute()
getAttribute()
removeAttribute()
getAttributeNames()

生命周期：

当前jsp页面被访问的时候创建，当前jsp访问结束的时候销毁。

作用范围：

当前jsp页面。

主要功能：

当前jsp页面内共享数据。