第六章 JSP技术

JSP技术是一种将Html静态模板内容（CSS、JS等），和Java程序相结合的技术，优点是：设计Web的用户界面程序非常简单，先按照静态网页来设计，等布局、颜色、美工等完成时，只需要在对应的位置嵌入java程序片，修改后缀名为\*\*\*.jsp，并且设置对应的编码，就好了。

弥补改善了Servlet程序在开发Web用户界面层的弊端。但是JSP程序本质上还是一个Servlet，在第一次运行某个Jsp程序是，web容器会将其翻译为Java文件，并且编译为class文件，就是一个Servlet程序。如果是Tomcat，则会存储在主目录的work文件夹中。

Jsp程序第一次运行可能会慢一些，第二次之后速度很快。具有Servlet程序常驻内存的特点。

JSP程序首先要了解内置对象，有九个。内置对象也称作隐式对象。首先作为对象，必须实例化后才能使用。内置对象就是不需要用户自己显示的实例化，Web容器启动时就已经实例化完成，用户直接使用的对象。

（1）out对象，本质就是Servlet中的PrintWriter，带缓冲的字符输出流。

6.3.2 include指令

Include指令的作用：可以实现JSP页面程序开发时的“插件式”设计，可以将一些统一样式，位置的内容，设计一份，其他页面中在相应位置进行保护即可。比如：web页面的logo区、copyright区等。优势在于修改、编辑时，一改全改。

Include指令又被成为，静态包含。静态包含在运行时相当于一个，复制被包含文件的内容，在包含指令出现的位置进行粘贴。最后在执行重新组合的文件。

6.4 JSP隐式对象

隐式对象，又称为内置对象，对象是必须实例化后才能使用的，对于程序员而言，使用对象两种方式：（1）自己显示的实例化（2）编程平台已经实例化的对象，直接使用。JSP中的隐式对象本质就是，Web容器在启动时已经实例化完成的对象，程序员直接使用即可。

JSP中有9大内置对象：

out对象，带缓冲的字符输出流，和PrintWriter类的对象功能类似，但是在jsp中做了优化，在JSp页面中就是用out，不要用PrintWriter，否则会产生乱序问题。

request对象，类型是HttpServletRequest，不多说了。

response对象，类型是HttpServletResponse，不多说了

config对象，类型是ServletConfig，功能和Servlet程序中的相应对象一样，在jsp中该对象几乎没用。

page对象，类型是Object，该对象其实就是JSP页面对应的“真身”Servlet程序中的this对象。基本上没用。

下面这四个对象本事大！重要！

这四个对象，称作域对象，或者属性范围对象，可以存储数据并且支持传递，区别在于能传多远多久。

application对象，类型是ServletContext，代表着当前web应用的存储范围。

session对象，类型是HttpSession，和Servlet中一样，不说了。

request对象，request不仅可以封装客户端提交的所有信息，并且是一个域对象，可以存储数据，传递时要注意请求不能改变！要使用请求转发。

pageContext对象，类型是PageContext，这个域对象的存值范围最小，仅仅在当前Jsp页面内有效，看似自己逗自己玩。这个域对象需要和EL表达式、JSTL标签等技术配合时，才是神。

pageContext对象还有一个绝活，就是有多个get\*\*\*方法，可以获取其他八个内置对象。

此外还有一个方法功能较强：setAttribute("str", str, PageContext.SESSION\_SCOPE);

一句话：pageContext一个顶8个。

6.5 JSP的动作元素

6.5.1 <jsp:include> 称作动态包含，运行测试发现，主jsp程序和被包含的jsp程序，分别转换、编译为不同的Servlet程序。运行效果的确是包含之后的结果，证明主jsp程序和被包含的jsp程序，分别运行出html的结果，然后整合在一起响应回客户端。

因为被包含的文件需要单独转换为Servlet，如果被包含的资源是jsp文件，则单独转换Servlet再将运行结果（html）和主文件的运行结果进行包含整合。如果被包含的文件是静态资源，直接将其运行结果和主文件的运行结果整合。

动态包含主文件和被包含文件中变量可以同名，不受影响！原因是，编译时分别转换为不同的Servlet，最后整合的是服务器运行的html结果。

6.5.2 <jsp:forward >动作元素

这个动作元素，实际上对应着Serlvet中的请求转发。该动作标签也被称作“服务器跳转”，功能和“请求转发”类似，但是有一点区别。请求转发跳转后，原来程序继续运行。而服务器跳转动作跳转之后，原程序终止不再执行。