Jsp 之客户端页面跳转

一、概述

一般来说页面跳转分为两类，一类是服务器端的页面跳转，还有一类是客户端的页面跳转。服务器端的页面跳转一般称为“转发”，客户端的页面跳转一般称为“重定向”。对于这两者的主要区别在下面通过列表的形式进行展示，由于不是这个文档的重点内容，因此就不加解释了。等到后面讲到 servlet 服务器编程的时候会再细说，这里你们就当了解一下。

服务器端跳转	客户端跳转
直接跳转，跳转前文件剩余代码不再执行	把跳转前文件的所有代码都执行完了以后再跳转
浏览器地址在跳转前后不会发生改变	浏览器地址在跳转前后很可能发生改变(例外：跳转到自己)
只能跳转到本服务器上的页面，不能跳转到其它服务器上的页面	可以跳转到其它服务器上的页面

要实现客户端的跳转一般有四种方式：

1、 使用超链接

2、 提交表单

3、 使用 JavaScript

4、 使用 response.setHeader 或者 response.sendRedirect

这里面第三种因为还没学到 JavaScript 因此暂时跳过，第四种一般结合 jsp 中的判断语句出现（不然就会出现用户什么都没动，页面加载

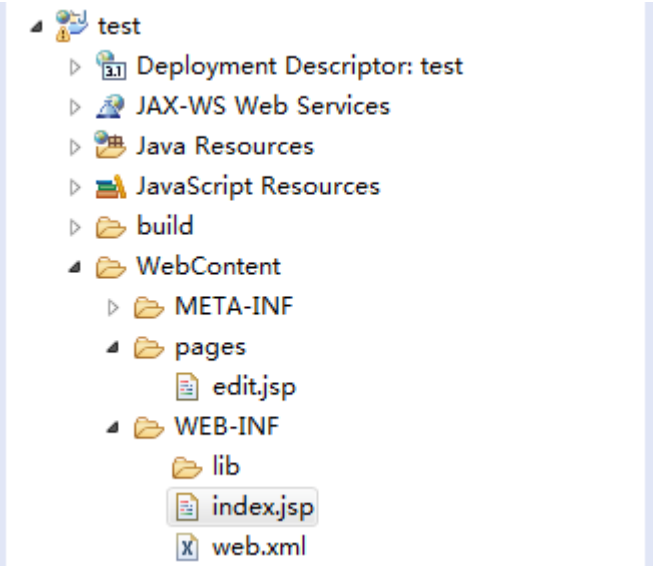
一结束立马跳转的神奇现象…)，而且用的也不如前三者多，因此这次重点介绍前两种方法。

二、超链接跳转

先说超链接的方法，这个方法最为简单，直接在 jsp 里面加一个 a 标签就可以了，然后在 url 那里写上绝对路径(比如说直接写上网址 www.baidu.com，这个自认为没什么难度)或者相对路径。

如果是相对路径要注意两点，第一跳转到的目标页面要放在 WebContent 文件夹下面，但是不能放在其下的 WEB-INF 文件夹下面(因为 WEB-INF 文件夹对于浏览器这种客户端来说是拒绝访问的，只有服务器自己才是可以访问的，因此客户端的跳转是跳转不到的)；第二要注意是相对路径里面的相对指的是相对地址栏显示的 url 的路径。

比如说我们在 eclipse 下面新建了一个文件夹 pages 使得目录变成如下的样子：



这时 edit.jsp 里面为

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="java.util.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.c
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <a href="edit.jsp"> edit.jsp </a>
    <a href="pages/edit.jsp"> edit.jsp </a>
</body>
</html>
```

这时怎么访问这个 jsp 文件呢。其实很简单，在 tomcat 启动以后直接地址栏打上 http://localhost:端口号/工程名/相对工程根目录的路径（比如说在这里为 http://localhost:8080/test/pages/edit.jsp）。

edit.jsp edit.jsp

这时候点击第一个超链接(上面写着相对路径 edit.jsp)，发现跳转到的依然是这个位置

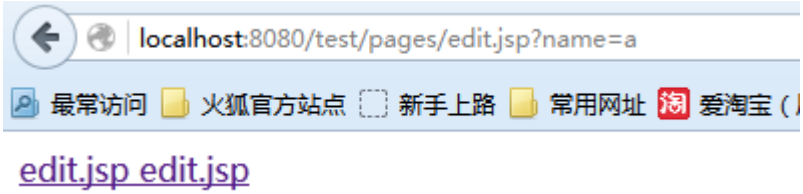
edit.jsp edit.jsp

这时候点击第二个超链接(上面写着相对路径 pages/edit.jsp)，发现跳转到的目录比原来多出一个 pages，自然找不到那个文件，从而出现 404 错误

这种方法麻烦在于参数传递。对于超链接来说最直观的参数传递就是参数直接写在地址栏里面。

```
<body>
    <a href="edit.jsp?name=a"> edit.jsp </a>
    <a href="pages/edit.jsp"> edit.jsp </a>
</body>
</html>
```

实际上？后面的都是参数，跟跳转页面的结果没有关系。这个参数可以通过 request.getParameter 得到。唯一不好的是地址栏莫名多了一些东西，而且如果参数是密码的话用这种方法非常危险。



三、表单提交跳转

对于表单提交跳转来说一般可用于跳转到本地的某个页面或者服务器 servlet，也可用于跳转到别的主机上的主页。下图中从上到下展示了跳转到相对路径，跳转到绝对路径，跳转到外部主页的三种方法。

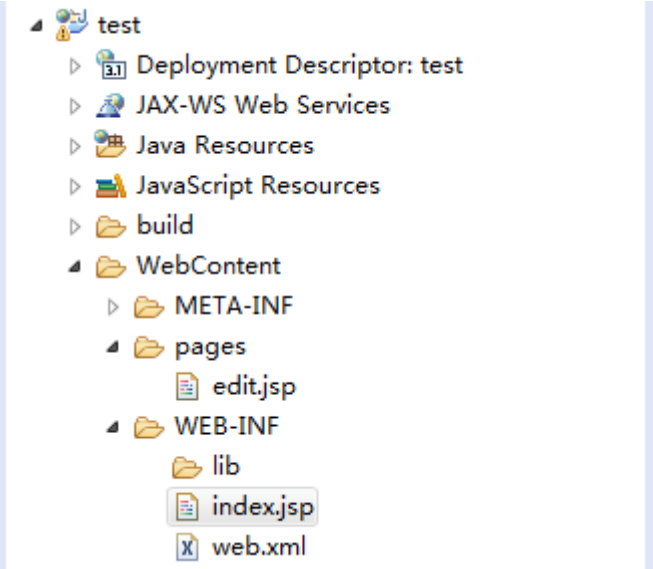
```
<body>
    <form action="file" method="post">
        <input name="save" type="submit" value="保存">
    </form>
</body>
```

```
<body>
    <form action="/file" method="post">
        <input name="save" type="submit" value="保存">
    </form>
</body>
```

```
<body>
    <form action="http://www.baidu.com" method="post">
        <input name="save" type="submit" value="保存">
    </form>
</body>
```

这时我们实现一个功能，欢迎页（首页）为 index.jsp。然后从欢迎页跳转到另一个页面。

假设我们的工程目录还是这样



这时我们的 web.xml 为

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xm
  <display-name>test</display-name>
  <welcome-file-list>
    <welcome-file>/WEB-INF/index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

需要跳转到 edit.jsp 最好的办法就还是写相对路径。这个相对路径和上面的一样，仍然是相对地址栏显示的 url 的路径。在这种情况下，由于欢迎页的 url 为 http://localhost:端口号/工程名（比如说在这里为 http://localhost:8080/test）。因此我们需要将 action 写成下面的形式。

```
<body>
    <form action="pages/edit.jsp" method="post">
        <input name="save" type="submit" value="保存">
    </form>
</body>
```

这时候就可以正常跳转过去了。

在这种情况下除了之前说到的直接地址栏中写参数的办法外，表单还会将 form 标签内部的所有 input 子标签的内容都传递过去。而且 method 里面还可以写上 post。这时候在地址栏那里显示的输入内容就会是加密之后的内容了；如果用的是 get，那么浏览器会将数据直接附在表单的 action URL 之后。这两者之间用问号进行分隔。这和之前直接在地址栏写参数的效果是一样的。