

java(Web)中相对路径，绝对路径问题总结

博客分类：

- [web前端](#)

转自：<http://www.iteye.com/topic/553749>

1.基本概念的理解

绝对路径：绝对路径就是你的主页上的文件或目录在硬盘上真正的路径，(URL和物理路径)例如：
C:\xyz\test.txt 代表了test.txt文件的绝对路径。<http://www.sun.com/index.htm>也代表了一个URL绝对路径。

相对路径：相对与某个基准目录的路径。包含Web的相对路径（HTML中的相对目录），例如：在Servlet中，"/"代表Web应用的跟目录。和物理路径的相对表示。例如："./" 代表当前目录，
"../"代表上级目录。这种类似的表示，也是属于相对路径。

另外关于URI，URL,URN等内容，请参考RFC相关文档标准。

RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax,
(<http://www.ietf.org/rfc/rfc2396.txt>)

2.关于JSP/Servlet中的相对路径和绝对路径。

2.1服务器端的地址

服务器端的相对地址指的是相对于你的web应用的地址，这个地址是在服务器端解析的（不同于html和javascript中的相对地址，他们是由客户端浏览器解析的）也就是说这时候在jsp和servlet中的相对地址应该是相对于你的web应用，即相对于<http://192.168.0.1/webapp/>的。

其用到的地方有：

forward：servlet中的request.getRequestDispatcher(address);这个address是

在服务器端解析的，所以，你要forward到a.jsp应该这么写：

request.getRequestDispatcher("/user/a.jsp")这个/相对于当前的web应用webapp，其绝对地址就是：<http://192.168.0.1/webapp/user/a.jsp>。

sendRedirect：在jsp中<%response.sendRedirect("/rtccp/user/a.jsp");%>

2.22、客户端的地址

所有的html页面中的相对地址都是相对于服务器根目录(<http://192.168.0.1/>)的，而不是(跟目录下的该Web应用的目录)<http://192.168.0.1/webapp/>的。

Html中的form表单的action属性的地址应该是相对于服务器根目录(<http://192.168.0.1/>)的，所以，如果提交到a.jsp为：action = "/webapp/user/a.jsp"或action = "

<%=request.getContextPath()%>"/user/a.jsp；

提交到servlet为action = "/webapp/handleservlet"

Javascript也是在客户端解析的，所以其相对路径和form表单一样。

因此，一般情况下，在JSP/HTML页面等引用的CSS,Javascript.Action等属性前面最好都加上<%=request.getContextPath()%>,以确保所引用的文件都属于Web应用中的目录。

另外，应该尽量避免使用类似".","/","../"等类似的相对该文件位置的相对路径，这样当文件移动时，很容易出问题。

站点根目录和css路径问题 (jsp是服务器端程序，地址是变化的，引用时一般用站点根目录的相对路径)

我们称类似这样的相对路径/test/.... 为相对于站点根目录 的相对路径。

当在jsp中引入css时，如果其相对路径相对于当前jsp文件的，而在一个和这个jsp的路径不一样的servlet中forward这个jsp时，就会发现这个css样式根本没有起作用。这是因为在servlet中转发时css的路径就是相对于这个servlet的相对路径而非jsp的路径了。所以这时候不能在jsp中用这样的路径：<link href="one.css" rel="stylesheet" type="text/css">或者<link href="../one.css" rel="stylesheet" type="text/css">类似href="one.css"和../one.css的html相对路径是相对于引用这个css的文件(a.jsp)的相对路径。而在servlet中转发时就是相对于这个servlet的相对路径了，因为**jsp路径和servlet路径是不一样的**，所以这样的引用肯定是出错的。

所以这个时候，要用站点根目录，就是相对于<http://192.168.0.1/>的目录，以 "/" 开头。

因此上述错误应更正为href=" /test/one.css" 类似的站点根目录的相对目录。这样在servlet转发后和jsp中都是相对于站点根目录的相对路径，就能正确使用所定义的css样式了。

3. JSP/Servlet中获得当前应用的相对路径和绝对路径

3.1 JSP中获得当前应用的相对路径和绝对路径

根目录所对应的绝对路径:request.getRequestURI()

文件的绝对路径 :application.getRealPath(request.getRequestURI());

当前web应用的绝对路径 :application.getRealPath("/");

取得请求文件的上层目录:new File(application.getRealPath(request.getRequestURI())).getParent()

3.2 Servlet中获得当前应用的相对路径和绝对路径

根目录所对应的绝对路径:request.getServletPath();

文件的绝对路径 :request.getSession().getServletContext().getRealPath(request.getRequestURI())

当前web应用的绝对路径 :servletConfig.getServletContext().getRealPath("/");

(ServletContext对象获得几种方式：

javax.servlet.http.HttpSession.getServletContext()

javax.servlet.jsp.PageContext.getServletContext()

javax.servlet.ServletConfig.getServletContext()

)

4.java 的Class中获得相对路径，绝对路径的方法

4.1单独的Java类中获得绝对路径

根据java.io.File的Doc文档，可知：

默认情况下new File("/")代表的目录为：System.getProperty("user.dir")。

一下程序获得执行类的当前路径

```
package org.cheng.file;
```

```
import java.io.File;
```

```
public class FileTest {  
    public static void main(String[] args) throws Exception {  
  
        System.out.println(Thread.currentThread().getContextClassLoader().getResource(""));  
  
        System.out.println(FileTest.class.getClassLoader().getResource(""));  
  
        System.out.println(ClassLoader.getSystemResource(""));  
        System.out.println(FileTest.class.getResource(""));  
        System.out.println(FileTest.class.getResource("/")); //Class文件所在路径  
        System.out.println(new File("/").getAbsolutePath());  
        System.out.println(System.getProperty("user.dir"));  
    }  
}
```

4.2服务器中的Java类获得当前路径（来自网络）

(1).Weblogic

WebApplication的系统文件根目录是你的weblogic安装所在根目录。

例如：如果你的weblogic安装在c:\bea\weblogic700.....

那么，你的文件根路径就是c:\.

所以，有两种方式能够让你访问你的服务器端的文件：

a.使用绝对路径：

比如将你的参数文件放在c:\yourconfig\yourconf.properties，
直接使用 new FileInputStream("yourconfig/yourconf.properties");

b.使用相对路径：

相对路径的根目录就是你的webapplication的根路径，即WEB-INF的上一级目录，将你的参数文件放

在你的webapp\yourconfig\yourconf.properties，

这样使用：

```
new FileInputStream("./yourconfig/yourconf.properties");
```

这两种方式均可，自己选择。

(2).Tomcat

在类中输出System.getProperty("user.dir");显示的是%Tomcat_Home%/bin

(3).Resin

不是你的JSP放的相对路径,是JSP引擎执行这个JSP编译成SERVLET

的路径为根.比如用新建文件法测试File f = new File("a.htm");

这个a.htm在resin的安装目录下

(4).如何读相对路径哪？

在Java文件中getResource或getResourceAsStream均可

例：getClass().getResourceAsStream(filePath);//filePath可以是"/filename",这里的/代表web

发布根路径下WEB-INF/classes

默认使用该方法的路径是：WEB-INF/classes。已经在Tomcat中测试。

页面跳转问题：

Forward 高, **Redirect** 低, 因为**Redirect** 的流程是这样的, request1 sent to server, server return back to client, 然后

request2 then sent to server. 但是**Forward** 仅在server side处理, 对client side 是透明的. 由于**Redirect** 有两次传输, 所以效率低.

范围:

由于对request.setAttribute() 来说, 它携带的对象生存范围只在request内, 所以**Redirect** 方式会导致request携带的对象丢失.

使用说明:

1. sendRedirect

servlet和jsp里面一样

```
response.sendRedirect();
```

2. include 这种也是上面提到的forward形式, request的值会保存

1) servlet里面

```
request.getRequestDispatcher( "jsp2.jsp" ).include(request, response);
```

2) jsp里面

```
<jsp:include page= "include.jsp" />
```

说明

页面会同时包含页面1和页面2的内容, 地址栏不变。

使用request.setAttribute的内容, 可以正常使用

3. forward

1) servlet里面

```
request.getRequestDispatcher( "jsp2.jsp" ).forward(request, response);
```

2) jsp里面

```
<jsp:forward page= "include.jsp" />
```

说明

页面会是页面2的内容, 地址栏不变

使用request.setAttribute的内容, 可以正常使用

3.JSP中用相对路径引用JS,CSS文件的三种情况

第一种情况 最常见的情况

一个tomcat上都跑多个工程, 用工程名来区分

因为我的URL是 :http://localhost/**工程名** /home/index.jsp

多了一个工程名,所以要加 `<%=request.getContextPath() %>`

如:`<script src=" <%=request.getContextPath() %> /home/test.js"> </script>`

写`<%=request.getContextPath() %>`太麻烦, 可以在每一个jsp文件顶部加入以下内容后,

Java代码

```
1. <%
2. String path = request.getContextPath();
3. String basePath =
   request.getScheme()+ "://" +request.getServerName()+":"+request.getServerPort()+path+"/" ;
4. %>
5. <base href="<%=basePath%">" >
```

就可直接使用 `<script src=" /home/test.js"> </script>`了

第二, 直接访问jsp文件

真实项目中, 一个tomcat上要是也跑了多个工程, 并用IP来区分

url 是这样的: `http://localhost/home/index.jsp` 注意这里,是直接访问JSP文件 (jsp文件不在web-inf下, 可以直接访问),不是servlet也不是struts。

test2.js 与index.jsp 放在同一文件夹下, 下面用相对路径来引入JS文件 是OK 的:

`<script src=test2.js> </script>`

index.jsp可以找到test2.js文件

第三种情况: servlet转发到jsp (jsp在web-inf下, 必须用servlet或action访问)

在第二情况的基础上, 一个tomcat上要是也跑了多个工程, 并用IP来区分。

我们访问的是servlet 或是struts的action, 再转发到 index.jsp

url 是: `http://localhost/***.do`

url 是: `http://localhost/index.action`

这里不是访问JSP文件了

下面是OK 的.

`<script src=/home/ test2.js> </script>`

test2.js 前面一定要有/home/

实际项目中, 访问 servlet或是struts的action最多, 所以这里要使用`<script src=/home/test2.js> </script>`。