

1. 把页面资源文件只能放在webroot下面,如 CSS,JS,image等.放在WEB-INF下引用不了.像jpetstore也是这样的.

2. 页面放在WEB-INF目录下面,这样可以限制访问,提高安全性.如JSP,html

3.只能用转向方式来访问WEB-INF目录下的JSP,不用采用重定向的方式请求该目录里面的任何资源.如图:index.jsp >>  main.jsp  
4.WEB-INF目录下文件访问资源文件时,可以忽略WEB-INF这一层目录.如main.jsp 要用css目录里的一个css文件.

    <link rel="stylesheet" type="text/css" href="**css/comm201005faa3.css**" />这样就行了,从客户端的地址可以看出来

    服务器转向main.jsp就是在webroot下面.所以main.jsp和css目录可以讲是同一级目录.  
5.WEB-INF/oa目录下访问images目录.怎么办呢.<img alt="" src="**images/instpage.gif**"></body>还是这这样.  
6.WEB-INF目录下的文件之间如何访问呢.如在main.jsp用<a href="**oa.do**">测试OA的路径</a>访问    像main.jsp有10处链接到WEB-INF目录下的其它页面.那就得有10个转向Action.这个可以用DispatchAction类加参数专门处理转向工作.

    注: 转向方式: 如struts-config文件中配置<forward name="success" path="/WEB-INF/main.jsp" />                     或 在Action中写request.getRequestDispatcher("/WEB-INF/main.jsp").forward(request, response);                 都是服务器读取了该页面内容,并发送到客户端.客户端的地址不变.内容跳转了  
    重定向方式: 如struts-config文件中配置<forward name="success" path="/WEB-INF/main.jsp"

                    redirect="true"/> 或在action中response.sendRedirect("/error.jsp");                     重定向的含义就是服务器把地址发给客户端,让客户端去访问.这种办法显然针对WEB-INF目录是无用功.