

1.若/交由浏览器来解析，代表当前web站点的根路径；例：

<http://localhost:8080/>

>超链接：`Java Web`
此时相当于：<http://localhost:8080/TestServlet> 此时是不能连接到servlet的，因为servlet不是直接部署tomcat的webapps目录下，正确的方式应该加上工程路径，即为：<http://localhost:8080/shipin/TestServlet>。

>表单中的action：`<form action="/login.jsp" >` 此时相当于：
<http://localhost:8080/login.jsp> 此时也是不能连接到servlet的，同上。

>请求重定向: `request.sendRedirect("/login.jsp")` 此时相当于：
<http://localhost:8080/login.jsp> 原理同上，浏览器直接更换网址

2.若/交由web服务器来解析代表当前web工程的根路径。例：`/shipin/`
(shipin即工程名)

>请求转发时：
`request.getRequestDispatcher("/chapter.jsp").forward(request, response);` 此时相当于：
<http://localhost/shipin/chapter.jsp> 若此时chapter.jsp在webContent或webRoot根目录下，可以连接，服务器内部转发

>web.xml 中映射servlet的访问路径时

`<servlet-mapping>`

`<servlet-name>getinRegisterServlet</servlet-name>`

```
<url-pattern>/getinRegister</url-pattern>
```

```
</servlet-mapping>
```

此时的getinRegisterServlet经过映射，实际路径为 <http://localhost:8080/shipin/getinRegister> 可以连接。

web应用中的“绝对路径”是指相对于工程根目录的路径

request.getContextPath()可获得

此时的路径在工程目录下，例：/shipin

例如:<a href="<%=request.getContextPath()%>/Test3Servlet">

里面有方法，全部转化为字符串后为"/shipin/Test3Servlet" 然后开始解析/

如上，这里/应该被解析为web站点的跟目录即<http://localhost:8080/>，解析完后即为完整的url，http://localhost:8080/shipin/Test3Servlet

classpath是指 WEB-INF文件夹下的classes目录

查看tomacat目录结构

这些xml配置文件都在WEB-INF\classes目录下，所以用classpath来得到这个下面的xml文件。