## 路径

### 1　与路径相关的操作

* 超链接
* 表单
* 转发
* 包含
* 重定向
* <url-pattern>
* ServletContext获取资源
* Class获取资源
* ClassLoader获取资源

### 2　客户端路径

超链接、表单、重定向都是客户端路径，客户端路径可以分为三种方式：

* 绝对路径；
* 以“/”开头的相对路径；
* 不以“/”开头的相对路径；

例如：http://localhost:8080/hello1/pages/a.html中的超链接和表单如下：

|  |
| --- |
| 绝对路径：<a href=*"http://localhost:8080/hello2/index.html"*>链接1</a>  客户端路径：<a href=*"/hello3/pages/index.html"*>链接2</a>  相对路径：<a href=*"index.html"*>链接3</a>  <hr/>  绝对路径：  <form action=*"http://localhost:8080/hello2/index.html"*>  <input type=*"submit"* value=*"表单1"*/>  </form>  客户端路径：  <form action=*"/hello2/index.html"*>  <input type=*"submit"* value=*"表单2"*/>  </form>  相对路径：  <form action=*"index.html"*>  <input type=*"submit"* value=*"表单3"*/>  </form> |

* 链接1和表单1：没什么可说的，它使用绝对路径；
* 链接2和表单2：以“/”开头，相对主机，与当前a.html的主机相同，即最终访问的页面为http://localhost:8080/hello2/index.html；
* 链接3和表单3：不以“/”开头，相对当前页面的路径，即a.html所属路径，即最终访问的路径为：http://localhost:8080/hello1/pages/index.html；

重定向1：

|  |
| --- |
| **public** **class** AServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  response.sendRedirect("/hello/index.html");  }  } |

　　假设访问AServlet的路径为：http://localhost:8080/hello/servlet/AServlet

　　因为路径以“/”开头，所以相对当前主机，即http://localhost:8080/hello/index.html。

重定向2：

|  |
| --- |
| **public** **class** AServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  response.sendRedirect("index.html");  }  } |

假设访问AServlet的路径为：http://localhost:8080/hello/servlet/AServlet

因为路径不以“/”开头，所以相对当前路径，即http://localhost:8080/hello/servlet/index.html

#### 2.1　建议使用“/”

强烈建议使用“/”开头的路径，这说明在页面中的超链接和表单都要以“/”开头，后面是当前应用的名称，再是访问路径：

<form action="/hello/servlet/AServlet">

</form>

<a href="/hello/b.html">链接</a>

其中/hello是当前应用名称，这也说明如果将来修改了应用名称，那么页面中的所有路径也要修改，这一点确实是个问题。这一问题的处理方案会在学习了JSP之后讲解！

在Servlet中的重定向也建议使用“/”开头。同理，也要给出应用的名称！例如：

|  |
| --- |
| response.sendRedirect("/hello/BServlet"); |

其中/hello是当前应用名，如果将来修改了应用名称，那么也要修改所有重定向的路径，这一问题的处理方案是使用request.getContextPath()来获取应用名称。

|  |
| --- |
| response.sendRedirect(request.getContextPath() + "/BServlet"); |

### 3　服务器端路径

服务器端路径必须是相对路径，不能是绝对路径。但相对路径有两种形式：

* 以“/”开头；
* 不以“/”开头；

其中请求转发、请求包含都是服务器端路径，服务器端路径与客户端路径的区别是：

* 客户端路径以“/”开头：相对当前主机；
* 服务器端路径以“/”开头：相对当前应用；

转发1：

|  |
| --- |
| **public** **class** AServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  request.getRequestDispatcher("/BServlet").forward(request, response);  }  } |

假设访问AServlet的路径为：http://localhost:8080/hello/servlet/AServlet

因为路径以“/”开头，所以相对当前应用，即http://localhost:8080/hello/BServlet。

转发2：

|  |
| --- |
| **public** **class** AServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  request.getRequestDispatcher("BServlet").forward(request, response);  }  } |

假设访问AServlet的路径为：http://localhost:8080/hello/servlet/AServlet

因为路径不以“/”开头，所以相对当前应用，即http://localhost:8080/hello/servlet/BServlet。

### 4　<url-pattern>路径

　　<url-pattern>必须使用“/”开头，并且相对的是当前应用。

### 5　ServletContext获取资源

必须是相对路径，可以“/”开头，也可以不使用“/”开头，但无论是否使用“/”开头都是相对当前应用路径。

例如在AServlet中获取资源，AServlet的路径路径为：http://localhost:8080/hello/servlet/AServlet：

|  |
| --- |
| **public** **class** AServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  String path1 = **this**.getServletContext().getRealPath("a.txt");  String path2 = **this**.getServletContext().getRealPath("/a.txt");  System.*out*.println(path1);  System.*out*.println(path2);  }  } |

path1和path2是相同的结果：http://localhost:8080/hello/a.txt