

Servlet 的层次结构

Servlet ---》GenericServlet ---》HttpServlet

HTTP 请求有很多种类型，常用的有四种：

GET 读取

POST 保存

PUT 修改

DELETE 删除

GenericServlet 实现 Servlet 接口，同时为它的子类屏蔽了不常用的方法，子类只需要重写 service 方法即可。

HttpServlet 继承 GenericServlet，根据请求类型进行分发处理，GET 进入 doGET 方法，POST 进入 doPOST 方法。

开发者自定义的 Servlet 类只需要继承 HttpServlet 即可，重新 doGET 和 doPOST。

```
package com.southwind.servlet;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/test")
public class TestServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        resp.getWriter().write("GET");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        resp.getWriter().write("POST");
    }
}
```

JSP 本质上就是一个 Servlet, JSP 主要负责与用户交互, 将最终的界面呈现给用户, HTML+JS+CSS+Java 的混合文件。

当服务器接收到一个后缀是 jsp 的请求时, 将该请求交给 JSP 引擎去处理, 每一个 JSP 页面第一次被访问的时候, JSP 引擎会将它翻译成一个 Servlet 文件, 再由 Web 容器调用 Servlet 完成响应。

单纯从开发的角度看, JSP 就是在 HTML 中嵌入 Java 程序。

具体的嵌入方式有 3 种:

1、JSP 脚本, 执行 Java 逻辑代码

```
<% Java代码 %>
```

2、JSP 声明: 定义 Java 方法

```
<%!  
    声明 Java 方法  
%>
```

3、JSP 表达式: 把 Java 对象直接输出到 HTML 页面中

```
<%=Java变量 %>
```

```
<%!  
    public String test(){  
        return "HelloWorld";  
    }  
%>  
  
<%  
String str = test();  
%>  
  
<%=str%>
```

JSP内置对象 9 个

- 1、request: 表示一次请求, HttpServletRequest。
- 2、response: 表示一次响应, HttpServletResponse。
- 3、pageContext: 页面上下文, 获取页面信息, PageContext。
- 4、session: 表示一次会话, 保存用户信息, HttpSession。

- 5、application：表示当前 Web 应用，全局对象，保存所有用户共享信息，ServletContext。
- 6、config：当前 JSP 对应的 Servlet 的 ServletConfig 对象，获取当前 Servlet 的信息。
- 7、out：向浏览器输出数据，JspWriter。
- 8、page：当前 JSP 对应的 Servlet 对象，Servlet。
- 9、exception：表示 JSP 页面发生的异常，Exception。

常用的是 request、response、session、application、pageContext

request 常用方法：

- 1、String getParameter(String key) 获取客户端传来的参数。
- 2、void setAttribute(String key,Object value) 通过键值对的形式保存数据。
- 3、Object getAttribute(String key) 通过 key 取出 value。
- 4、RequestDispatcher getRequestDispatcher(String path) 返回一个 RequestDispatcher 对象，该对象的 forward 方法用于请求转发。
- 5、String[] getParameterValues() 获取客户端传来的多个同名参数。
- 6、void setCharacterEncoding(String charset) 指定每个请求的编码。

HTTP 请求状态码

200：正常

404：资源找不到

400：请求类型不匹配

500：Java 程序抛出异常

response 常用方法：

- 1、sendRedirect(String path) 重定向，页面之间的跳转。

转发 getRequestDispatcher 和重定向 sendRedirect 的区别：

转发是将同一个请求传给下一个页面，重定向是创建一个新的请求传给下一个页面，之前的请求结束生命周期。

转发：同一个请求在服务器之间传递，地址栏不变，也叫服务器跳转。

重定向：由客户端发送一次新的请求来访问跳转后的目标资源，地址栏改变，也叫客户端跳转。

如果两个页面之间需要通过 request 来传值，则必须使用转发，不能使用重定向。

用户登录，如果用户名和密码正确，则跳转到首页（转发），并且展示用户名，否则重新回到登陆页面（重定向）。