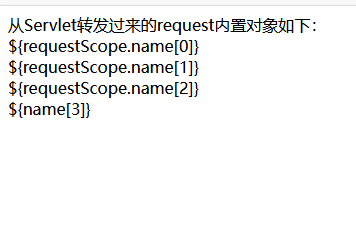
<%@ **page** language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  
<body>  
从Servlet转发过来的request内置对象如下：<br>  
${requestScope.name[0]}<br>  
${requestScope.name[1]}<br>  
${requestScope.name[2]}<br>  
${name[3]}<br>  
</body>  
</html>



JSP页面与EL表达式分析

核心内容解析

1.EL 表达式核心功能

简化数据访问：

使用 ${ } 替代传统 JSP 脚本（如 <%=request.getAttribute()%>），消除 Java 代码嵌入。

自动作用域查找：

${name[3]} 省略作用域前缀时，按 page→request→session→application 顺序自动搜索属性。

2.EL 隐含对象使用

显式作用域指定：

${requestScope.name[0]} 直接从 request 作用域获取数据，避免歧义。

数组/集合访问：

使用下标 [n] 访问数组或 List 元素（如 name[0]）。

3.数据来源机制

依赖 Servlet 转发时设置的属性：

java

request.setAttribute("name", nameArray); // 转发前存入数组

request.getRequestDispatcher("result.jsp").forward(request, response);

实验心得：EL 隐含对象深度解析

1. EL 隐含对象分类

隐含对象 作用 JSP 等效代码

requestScope 访问 request 作用域属性 request.getAttribute()

sessionScope 访问 session 作用域属性 session.getAttribute()

applicationScope 访问 application 作用域属性 application.getAttribute()

param 获取请求参数（单值） request.getParameter()

paramValues 获取请求参数（多值，如复选框） request.getParameterValues()

header 获取 HTTP 请求头 request.getHeader()

2. 关键技巧演示

安全访问嵌套属性：

${user?.address?.city} 避免空指针（若 user 为 null 不抛异常）。

条件渲染：

html

<div>${not empty sessionScope.user ? "已登录" : "未登录"}</div>

隐式参数获取：

${param.id} 等价于 <%=request.getParameter("id")%>。

3. 优势总结

代码简洁性：

传统 JSP 脚本：<% String name = (String)request.getAttribute("name"); %>

EL 表达式：${name}

类型自动转换：

数字、布尔值等自动转换，无需手动处理。

防 XSS 攻击：

默认对 ${ } 输出进行 HTML 转义（> 转义为 &gt;）。

4. 注意事项

作用域优先级陷阱：

若不同作用域有同名属性，${name} 可能返回非预期对象（如 session 优先于 request）。

解决方案：显式指定作用域 ${requestScope.name}。

EL 功能开关：

在 JSP 2.0+ 中默认启用，旧版本需在 web.xml 启用：

xml

<jsp-config>

<jsp-property-group>

<url-pattern>\*.jsp</url-pattern>

<el-ignored>false</el-ignored>

</jsp-property-group>

</jsp-config>

5. 扩展应用场景

访问 Map 数据：

${map.key} 或 ${map["key-with-dash"]}。

调用 JavaBean 方法：

${user.getName()} 可简写为 ${user.name}（需符合 JavaBean 规范）。

集合流操作 (JSTL+EL)：

jsp

<c:forEach items="${books}" var="book">

${book.price > 50 ? "高价" : "平价"}

</c:forEach>

总结：该页面是 EL 表达式应用的典型示例，展示了如何优雅访问转发数据。EL 通过隐含对象和作用域规则，将复杂的数据访问简化为声明式语法，大幅提升 JSP 可维护性。结合 JSTL 可构建无脚本的动态页面，符合现代 Java Web 开发最佳实践。