# Servlet

## Servlet初始化

@WebServlet( 定义为servlet

urlPatterns = { "/scope" }, 指定serrlet相应路径

initParams = { @WebInitParam(name= "initnum", value = "5") },

初始化参数 (参数名 参数值 )

loadOnStartup = 1 不为0/null 随项目启动此servlet

)

**public** **void** init(ServletConfig config) **throws** ServletException {

**int** initnum = Integer.*parseInt*(config.getInitParameter("initnum"));

}

当有2个init方法时，默认执行有参数的init()方法

doGet / doPost 都属于doService 当有Service()方法时，默认调用Service()方法 其他doXX方法不执行

## 编码问题

Get 设置 在tomcat\_home\conf\server.xml 中的Connector元素中设置URIEncoding属性为合适的字符编码 URIEncoding="UTF-8"

Post请求设置request.setCharacterEncoding("UTF-8");

## JSP常用内置对象

### Out( PrintWriter ) /使用<%= %>代替

### Request(HttpServletRequest)

### Response(HttpServletResponse)

### Session(HttpSession)

### Application(ServletContext)

## JSP原理 ==Servlet

<% %> JAVA代码 <%! %>定义全局变量 !定义为全局变量

JSP引入JAVA包 <%@ page import=*"java.util.\*"*%>

## JSP跳转方式(转发/重定向)

重定向 response.sendRedirect("jsp/01.jsp"); 当所有语句执行完毕才会重定向，加入return ，可立即执行。

跳转不能传递中文参数(客户端跳转)

转发

request.getRequestDispatcher("jsp/01.jsp").forward(request, response); (服务器端跳转 )

## Servlet路径

request.getContextPath(); 获得上下文路径，即项目名称级别路径，方便使用绝对路径

## JSP页面包含

动态包含

<%@ include file=*"/include/top.jsp"* %> 使用绝对路径包含时，会自动添加上下文路劲 ，不需要手动添加上下文路径

静态包含

<jsp:include page=*"/include/top2.jsp"* />

区别，静态包含，先将JSP进行编译为HTML后，包含，动态包含，先包含JSP，在被导入的页面进行编译,动态包含可能发生 多次定义 同名变量

## JSP内置对象的作用域

PageContext 当前页面存在，发送跳转不存在

Request 当前页面存在，服务器跳转存在，客户端跳转不存在

Session 只有关闭浏览器才丢失，每个用户单独享有

Application 只有服务重启后才丢失，整个项目共享

能用作用域小的就用小的

## 过滤器

**implements** Filter 实现该类，定义为Filter

@WebFilter(

urlPatterns="/\*", 指定过滤的地址

initParams={@WebInitParam(name="charset",value="UTF-8")})

指定初始化参数

doFilter(ServletRequest request, ServletResponse response,

FilterChain chain){

request.setCharacterEncoding(charset);

chain.doFilter(request, response); }

拦截器链 需doFilter,以执行后续Filter 或 Servlet

**public** **void** init(FilterConfig cfg) **throws** ServletException {

String c = cfg.getInitParameter("charset");

**if** (**null** != c && "" != c.trim()) {

charset = c;

**return**;

}

charset = "UTF-8";

}

## Servlet路径跳转问题

### Web.xml对于Servlet 的url的定义 必须以/开头 ，/表示项目的根路径

### JSP页面的<a>请求路径 / 表示服务器的根路径，没有项目的根路径，若以/开头必须 添加项目的根路径 ContextPath

### Servlet重定向的路径，相当于<a>标签的客户端跳转 同上 response.sendRedirect(“……..”)

### Servlet 转发，服务器端跳转路径 /表示项目的根路径，request.getRequestDispatcher(“……”).forward(req,res)