①在页面输入中文，请求页面后不出现乱码：

contentType="text/html;charset=UTF-8"和pageEncoding="UTF-8"

保持一致UTF-8，还需保证浏览器显示的字符编码也为UTF-8

②获取中文属性不出现乱码，对于post请求 ：在获取请求参数之前先调用setCharacterEncoding("UTF-8")方法设置字符编码

读取：

 request.setCharacterEncoding("utf-8");

写出：

  response.setContentType("text/html;charset=utf-8");

  response.setCharacterEncoding("utf-8");

**两个必须同时设置**

**做一个filter**

**public** **class** EncodingFilter **extends** HttpFilter {

**private** String encoding;

**protected** **void** init() {

// 获取Context中的属性encoding

encoding = getFilterConfig().getServletContext().getInitParameter("encoding");

}

@Override

**public** **void** doFilter(HttpServletRequest request, HttpServletResponse response, FilterChain chain)

**throws** IOException, ServletException {

**if** (encoding == **null**) {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

chain.doFilter(request, response);

**return**;

}

request.setCharacterEncoding(encoding);

response.setContentType("text/html;charset="+encoding);

response.setCharacterEncoding(encoding);

chain.doFilter(request, response);

}

}

**Web.xml中**

<context-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</context-param>

<filter>

<filter-name>EncodingFilter</filter-name>

<filter-class>filter.EncodingFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>EncodingFilter</filter-name>

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

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>