

# 问题概述

## post请求中文乱码

### 1. 产生原因:

，页面定义编码我们常采用utf-8的格式。数据传输的时候表单中的参数值对是通过request包发送给服务器，此时浏览器会根据网页默认的Contentype("text/html; charset=iso8859-1")中指定的编码进行对表单中的数据进行编码，然后发给服务器。从而产生乱码 2. 解决在web.xml中加载过滤器，spring去做了代理，只需要配置到参数即可。将请求编译的字符集设置成utf-8的格式。从而避免中文乱码问题。

```
<filter>
<filter-name>CharacterEncodingFilter</filter-name>
<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
<init-param>
<param-name>encoding</param-name>
<param-value>utf-8</param-value>
</init-param>
</filter>
<filter-mapping>
<filter-name>CharacterEncodingFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
```

## get乱码

### 1. 产生原因:

- 请求数据附加到URL后面作为参数，这样依赖乱码就会很容易出现，因为数据name和value很有可能就是传递的为非ASCII码。当URL拼接后，浏览器对其进行encode，然后发送到服务器。在这个编译的过程中URL encode的字符一般都是非ASCII码字符，所以出现乱码主要是URL中附加了中文或特殊字符，而编码方式是由浏览器决定的，不同的浏览器和同一浏览器的不同设置影响了URL的编码，所以为了避免我们不需要的编码，我们可以通过java代码或javascript代码统一进行控制。也就是页面的编写定义为utf-8的字符集。
- 请求到了服务器之后，由浏览器（utf-8）的编译到达tomcat后，tomcat内核的字符解析器是iso8859-1的解析器，最终到达我们的后端代码自然出现了中文乱码。

### 2. 对于get请求中文参数出现乱码解决方法有两个:

- 修改tomcat配置文件添加编码与工程编码一致，如下:

```
<Connector URIEncoding="utf-8" connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443"/>
```

- 另外一种方法对参数进行重新编码:

```
String userName new
String(request.getParamter("userName").getBytes("ISO8859-1"),"utf-8")
```

ISO8859-1是tomcat默认编码，需要将tomcat编码后的内容按utf-8编码

## springMVC与struts的区别

- springmvc的入口是一个servlet即前端控制器，而struts入口是一个filter过滤器。
- springmvc是基于方法开发(一个url对应一个方法)，请求参数传递到方法的形参，可以设计为单例或多例(建议单例)，struts是基于类开发，传递参数是通过类的属性，只能设计为多例。
- Struts采用值栈存储请求和响应的数据，通过OGNL存取数据，springmvc通过参数解析器是将request请求内容解析，并给方法形参赋值，将数据和视图封装成ModelAndView对象，最后又将ModelAndView中的模型数据通过request域传输到页面。Jsp视图解析器默认使用jstl。

完