# 编写编码控制的filter

public class SetCharacterEncodingFilter implements Filter { //要实现Filter接口

 //存储编码格式信息  
 private String encode = null;

 public void destroy(){  
 }  
 public void doFilter(ServletRequest req,ServletResponse resp,FilterChain chain)  
   throws IOException,ServletException{  
  //转换  
  HttpServletRequest request = (HttpServletRequest)req;  
  HttpServletResponse response = (HttpServletResponse)resp;  
  /\*  
   \* 判断在web.xml文件中是否配置了编码格式的信息  
   \* 如果为空，则设置编码格式为配置文件中的编码格式  
   \* 否则编码格式设置为UTF-8  
   \*/  
  if(this.encode != null && !this.encode.equals("")){  
    request.setCharacterEncoding(this.encode);  
    response.setCharacterEncoding(this.encode);  
  }else{  
    request.setCharacterEncoding("UTF-8");  
    response.setCharacterEncoding("UTF-8");  
  }  
  /\*  
   \* 使用doFilter方法调用链中的下一个过滤器或目标资源（servlet或JSP页面）。  
   \* chain.doFilter处理过滤器的其余部分（如果有的话），最终处理请求的servlet或JSP页面。  
   \*/  
  chain.doFilter(request, response);  
 }  
 public void init(FilterConfig config) throws ServletException{  
  //获取在web.xml文件中配置了的编码格式的信息  
  this.encode = config.getInitParameter("encode");  
 }  
}

# 在web.xml里配置

<filter>  
 <filter-name>encodingFilter</filter-name>  
 <filter-class>com.nubia.framework.core.CoreFilter</filter-class>  
 <init-param>  
 <param-name>encoding</param-name>  
 <param-value>utf-8</param-value>  
 </init-param>  
</filter>  
<filter-mapping>  
 <filter-name>encodingFilter</filter-name>  
 <url-pattern>/\*</url-pattern>  
</filter-mapping>