JSP 以及 Servlet 中文乱码问题

关于在 JavaWeb 中经常会出现乱码的状况，下面总结一下会出现乱码的情况：

一、JSP 乱码

这种是最常见的，设置编码的位置位于 JSP 的第一行，如果在 Eclipse 中新建一个 JSP 默认是下面这种：

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

可以看到它默认的页面编码和传输编码都是 ISO-8859-1,这是用于欧洲国家的编码。

如果想要支持中文，可以使用 UTF-8、GB2312、GBK 等，其中 UTF-8 是国际化的，哪个国家的都支持，所以推荐使用这个。

- 再来说说上面涉及到编码的两个地方：charset 和 pageEncoding

  charset 是指服务器发往客户端展现时的编码；

  pageEncoding 用于设置 JSP 页面本身的编码。

  JSP 在部署后提供给用户使用，会经过三个阶段：

1、 JSP 生成 java 文件：这个阶段会使用 pageEncoding 所定义的编码格式进行转换

2 、java 文件生成 class 文件 :这个阶段由服务器 tomcat 自动使用 utf-8 编码把 java 文件转换成字节码 class 文件

3、 通过读取 class 文件展现给用户：这个阶段由 tomcat 服务器获取字节码内容，通过使用 contentType 所定义的编码格式展现给用户。

这样设置好 JSP 中的第一行代码，就可以保证基本的 JSP 展现没有乱码了 !需要注意：

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

charset、pageEncoding 以及 contentType 三者的编码格式需一致。

以下是示例代码：

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http:/
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>成功页面</title>
</head>
```

二、request 中文乱码

有时候在做 jsp 逻辑处理时，比如提交表单，从前台注册的页面提交了一部分的数据，但是后面处理的 JSP 页面通过 request.getParameter 调用时，获取到的是一堆乱码。这是因为虽然前面 JSP 设置了编码格式，却没有在当前的 JSP 中设置读取数据的编码格式。

1. 使用下面的代码，就可以是设置 request 获取请求内容的数据编码：

   request.setCharacterEncoding("utf-8");

   request.setCharacterEncoding()作用以及使用注意事项：

   作用：设置从 request 中取得的值或从数据库中取出的值的编码方式；也就是说该方法用来指定对浏览器发送来的数据进行重新编码（或者称为解码）时，使用的编码。

   使用注意事项：

   1）在执行 setCharacterEncoding()之前，不能执行任何 getParameter()操作。

   2） 通过 setCharacterEncoding 设置的编码方式只对 POST 方式提交的表单有效，对 GET 方式无效。也就是说只对 method="post"方式有效。

   分析原因：

1） 在执行第一个 getParameter()的时候，java 将会按照编码分析所有的提交内容，而后续的 getParameter()不再进行分析，所以 setCharacterEncoding()无效。

2） 对于 GET 方式提交的表单，提交的内容在 URL 中，一开始就已经按照编码分析提交内容，setCharacterEncoding()自然就无效。

2. 需要注意的是，这种方式对 URL 传参这种 JSP 请求是没有作用的。比如：

url test request(zh)

这种情况仍然会出现乱码，这种 URL 传参的方式，只能修改服务器 tomcat 的传输编码格式。修改 tomcat 安装文件 apache-tomcat-8.5.20\conf 目录下的 server.xml

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

添加 URIEncoding="UTF-8"，就可以处理 URL 传递参数造成的中文乱码问题了。

三、response 输出中文乱码

response 输出页面元素内容时，也会出现乱码。

使用下面代码就可以设置 response 输出的编码格式：

response.setCharacterEncoding();

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

response.setCharacterEncoding()作用：

作用：指定对服务器响应进行重新编码的编码。服务器在将数据发送到浏览器前，对数据进行重新编码时，使用的就是该编码。

四、Servlet 中文乱码

服务器获取客户端传递过来的数据出现乱码的问题(即使用 get 获取 post 向服务器发送请求时出现乱码)：

1. post 请求乱码处理 ：

用户在表单中填写的内容在 http 体中被提交给 Servlet。当我们在表单中输入中文时，servlet 的 request 的编码与客户端不一致，则服务器无法解析，则会出现乱码。**解决方法**是改变 http 请求体中的字符编码(对于 get 无效，因为 get 提交在请求头中)

改变 http 请求体中的字符编码为 UTF8 : request.setCharacterEncoding("UTF-8");

2. get 请求乱码处理

1）URL 的参数没有使用编码(即使用了 ISO-8859-1 等)，则在服务器端获取 get 的参数的时候使用 String 一个可以指定编码的构造函数。

String username = request.getParameter("username");

String resultName = new String(username.getBytes("ISO-8859-1"),"utf-8");

2）有时提交的一些信息在地址栏显示的是“%2C%C6%CC%C6”的字样，其实这就是对 URL 进行了如下的编码。

java 有两个类用来修改编码：

进行编码：URLEncoder.encode(String s, "UTF-8")

进行解码：URLDecoder.decode(String s, "UTF-8");

这样就可以得到传递过来的中文参数了 ,许多网站用的都是这种方式解决中参数的。

使用这个类把 URL 修改为:

http://127.0.0.1:8080/test?username=<%=URLEncoder.encode("乱码文字","UTF8")%>

这样服务器获取 username 这个参数使用:

String username =

URLDecoder.decode(request.getParameter("username"),"UTF8");

以下是示例代码：

示例代码展示了解决乱码的几种方式，是综合性的；在实际应用中，可能使用其中某一种方式就可以解决。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>登录页面</title>
</head>
<body>
        <form action="/reg_login/LoginServlet" method="post">
            <table>
                <tr>
                    <td class="td1">用户名</td>
                    <td><input type="text" class="input" name="username"></td>
                </tr>
                <div>
                    <input type="submit" value="提交" id="login-btn">
                </div>
            </table>
        </form>
</body>
</html>
```

图示是 Servlet 中，解决乱码的方式：

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    // 接收数据：
    // 设置request请求参数的编码方式
    request.setCharacterEncoding("UTF-8");
    // 设置response输出的编码方式
    response.setCharacterEncoding("UTF-8");
    response.setContentType("text/html;charset=UTF-8");

    // 获取字符串并设置获取字符串的编码
    String username = new String(request.getParameter("username").getBytes("ISO-8859-1"), "UTF-8")
    System.out.println(username);

    request.setAttribute("username", username);
    request.getRequestDispatcher("/success.jsp").forward(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    doGet(request, response);
}
```

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http:/
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <%  String name = request.getParameter("username");
        out.print(name);
    %>
</body>
</html>
```