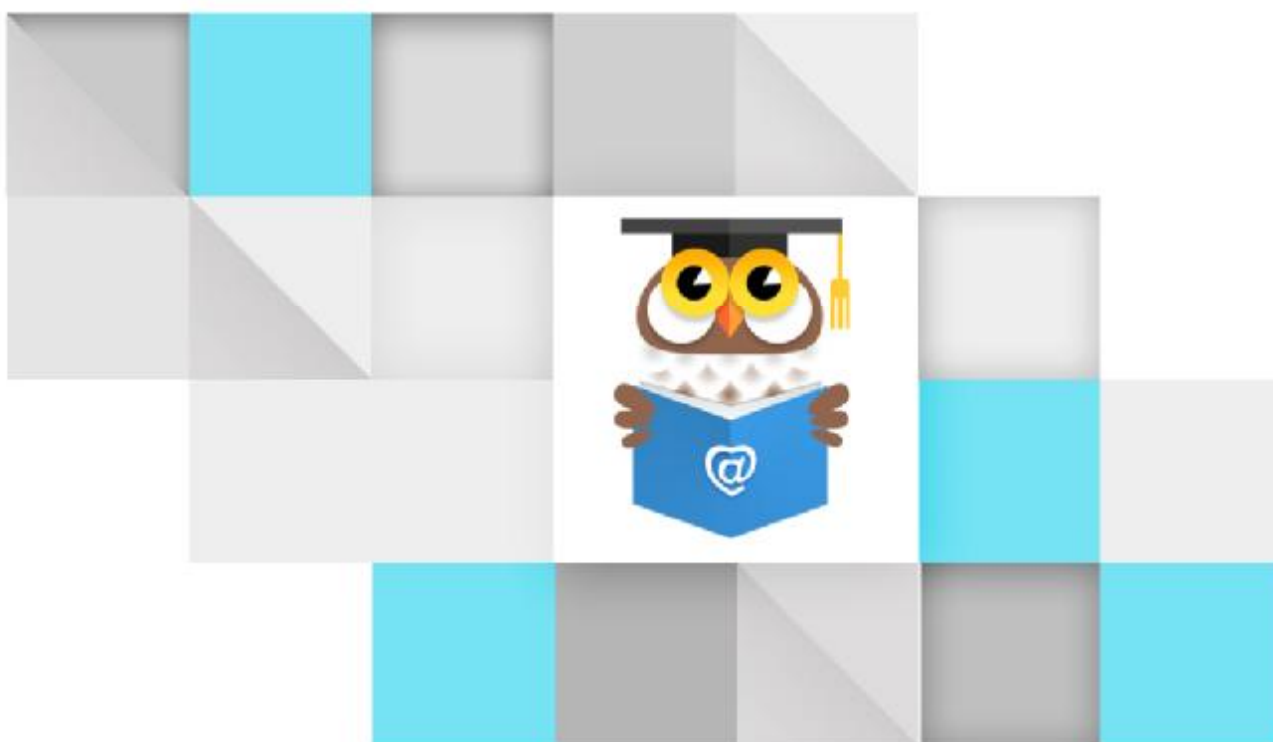


去哪玩旅游网（动态 WEB）

教学指导手册

PRJ-WTP-JEE-016 – 乱码过滤器



Campus Solution Group

目 录

一、场景说明	1
1、完成效果	1
2、业务描述	1
3、实现思路	1
4、核心组件	1
二、实训技能	2
1、重点演练	2
2、相关技能	2
3、相关知识点	3
4、前置条件	3
5、搭建环境	3
三、场景任务	4
任务 1、创建并配置 Filter 组件	4
任务 2、处理 POST 类型请求中的乱码	7
任务 3、处理 GET 类型请求中的乱码	9

一、场景说明

1、完成效果



图 1-1-1

2、业务描述

本场景主要用于确保发往《去哪玩》旅游网请求中，含中文字符的数据不出现乱码。

2-1. 利用过滤器Filter技术，拦截所有发往《去哪玩》旅游网的请求。

2-2. 确保所有POST类型的请求数据中，中文字符不出现乱码。

2-3. 确保所有GET类型的请求数据中，中文字符不出现乱码。

3、实现思路

3-1. 本场景建议将“**处理中文字符乱码**”的业务分为三个任务依次实现：

3-1.1. 任务1. 创建负责清除乱码的过滤器（Filter）组件：**EncodingFilter**。

3-1.2. 任务2. 确保所有客户端发送的**POST请求数据中**，中文字符无乱码。

3-1.3. 任务3. 确保所有客户端发送的**GET请求数据中**，中文字符无乱码。

4、核心组件

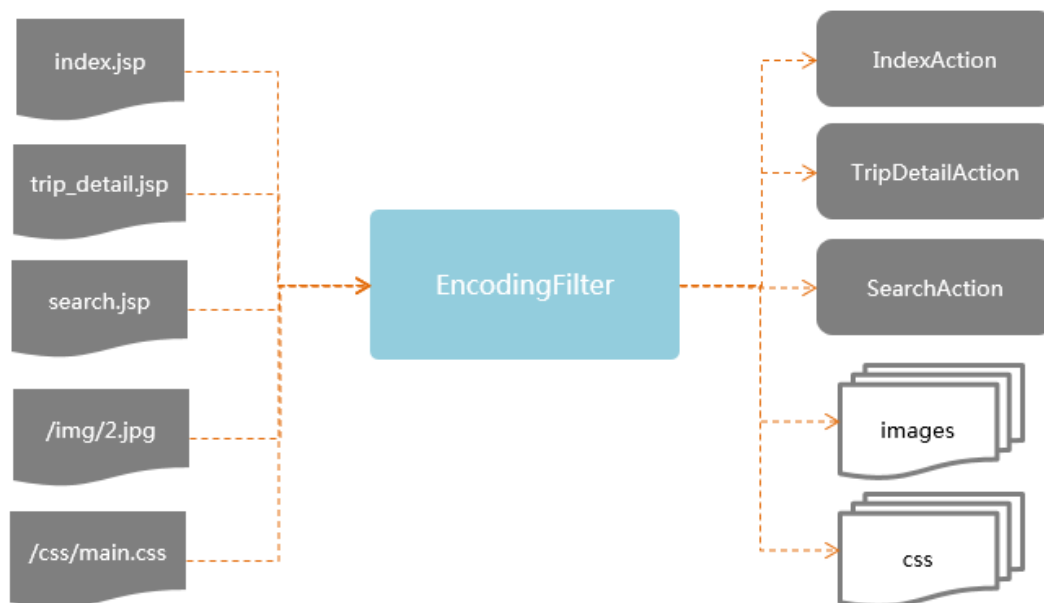


图 1-4-1

4-1. EncodingFilter (需实现) :

4-1.1. 字符集过滤器，负责拦截所有客户端请求。

4-1.2. 当前场景，该过滤器主要负责处理中文字符乱码的问题。

二、实训技能

1、重点演练

1-1. 创建、配置、使用Filter组件。

1-2. 通过FilterConfig对象，读取web.xml文件中的配置信息。

1-3. 理解POST请求类型中，中文字符出现乱码的原因。

1-4. 理解GET请求类型中，中文字符出现乱码的原因。

2、相关技能

I Filter 技术应用

3、相关知识点

- I Filter 接口
- I Filter 配置
- I Filter 与 Servlet

4、前置条件

4-1. 前置场景：PRJ-WTP-JEE-010 检索页信息显示

4-2. 已学技能：

4-2.1. Java开发工具（Eclipse）。

4-2.2. Web容器（Tomcat）。

4-2.1. Servlet应用技术。

4-2.2. Filter应用技术。

5、搭建环境

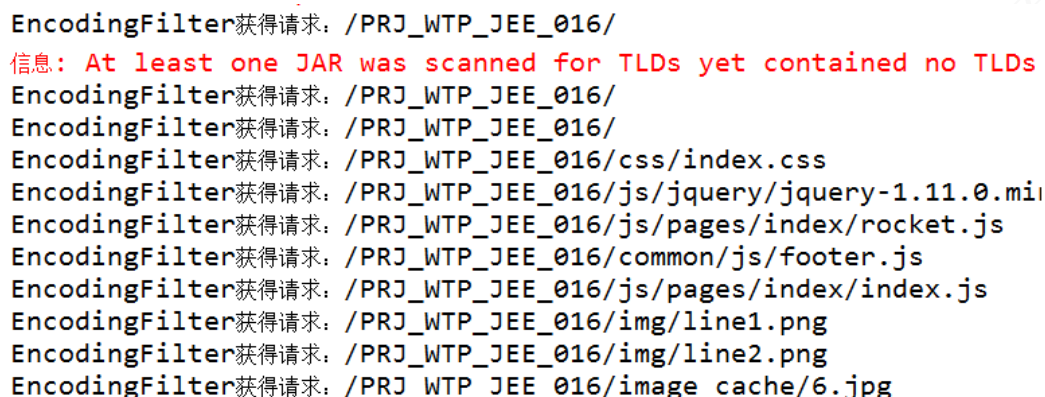


三、场景任务

任务 1、创建并配置 Filter 组件

1. 任务说明：

1-1. 完成效果：



```
EncodingFilter获得请求: /PRJ_WTP_JEE_016/
信息: At least one JAR was scanned for TLDs yet contained no TLDs
EncodingFilter获得请求: /PRJ_WTP_JEE_016/
EncodingFilter获得请求: /PRJ_WTP_JEE_016/
EncodingFilter获得请求: /PRJ_WTP_JEE_016/css/index.css
EncodingFilter获得请求: /PRJ_WTP_JEE_016/js/jquery/jquery-1.11.0.min.js
EncodingFilter获得请求: /PRJ_WTP_JEE_016/js/pages/index/rocket.js
EncodingFilter获得请求: /PRJ_WTP_JEE_016/common/js/footer.js
EncodingFilter获得请求: /PRJ_WTP_JEE_016/js/pages/index/index.js
EncodingFilter获得请求: /PRJ_WTP_JEE_016/img/line1.png
EncodingFilter获得请求: /PRJ_WTP_JEE_016/img/line2.png
EncodingFilter获得请求: /PRJ_WTP_JEE_016/image_cache/6.jpg
```

图 3-1-2

1-2. 任务目标：

1-2.1. 创建负责清除乱码的过滤器（Filter）组件。

1-3. 任务要求：

1-3.1. 创建负责清除乱码的过滤器（Filter）组件：

- 1）限制1. Filter取名：**EncodingFilter**。
- 2）限制2. **EncodingFilter**创建于**campsg.qunawan.filter**包中。
- 3）限制3. 允许通过**任意类型**URL（**/***）访问该Filter。

1-3.2. 当**EncodingFilter**获取请求后，在控制台输出：

- 1）Filter创建时，输出 **“EncodingFilter被创建”**。
- 2）Filter接收到请求时，输出：**“EncodingFilter获得请求：xxxxxx”**。

2. 实现思路：

2-1. 创建清除乱码的过滤器：EncodingFilter：

2-1.1. 在campsg.qunawan.filter包中创建Class类，命名：EncodingFilter。

2-1.2. 使EncodingFilter实现javax.servlet.Filter接口。

2-1.3. 重写javax.servlet.Filter接口中的init、doFilter、destory方法。

2-2. 配置EncodingFilter：

2-2.1. 在web.xml中配置EncodingFilter。

2-2.2. 将EncodingFilter的url-pattern设置为：/*

2-3. 使用EncodingFilter：

2-3.1. 在EncodingFilter的init方法中，打印：“EncodingFilter被创建”。

2-3.2. 在EncodingFilter的doFilter方法中，获取请求URL连接。

2-3.3. 随后打印：“EncodingFilter获得请求：” 拼接获取的URL连接。

3. 推荐步骤：

3-1. 按实现思路，创建名为EncodingFilter的过滤器Filter。

3-2. 按实现思路，配置名为EncodingFilter的过滤器Filter。

3-3. 使用EncodingFilter：

3-3.1. 在init方法中，打印：“EncodingFilter被创建”。

3-3.2. 在doFilter方法中，获取请求URL并输出：

+ 提示：

1) 在Filter类的doFilter方法获取请求URL，需依次执行以下步骤：

1-1. 将ServletRequest对象强制类型转换成：HttpServletRequest。

1-2. 通过HttpServletRequest类的getRequestURI方法，获取请求URL

1-3. 代码如下所示：

```
HttpServletRequest req = (HttpServletRequest) request;  
System.out.println("EncodingFilter获得请求: " + req.getRequestURI());
```

说明1：过滤器不仅可以处理HTTP协议的请求，还可处理诸如：FTP、SOAP等网络协议的请求。

说明2：当前场景使用HTTP通讯协议，因此获取URL时，需通过强转获取HttpServletRequest。

3-3.3. 最后，执行过滤链（FilterChain）的doFilter方法，运行下一Web组件。

4. 验证与测试：

4-1. 运行项目工程。

4-2. 使用Chrome浏览器访问该项目（必须）。

4-3. 在浏览器地址栏输入 http://localhost/PRJ_WTP_JEE_016/ 访问首页。

4-4. 观察控制台，应见【任务完成效果截图】。

任务 2、处理 POST 类型请求中的乱码

1. 任务说明：

1-1. 完成效果：

EncodingFilter获得请求: /PRJ_WTP_JEE_016/image_cache/39.jpg
EncodingFilter获得请求: /PRJ_WTP_JEE_016/image_cache/45.jpg
EncodingFilter获得请求: /PRJ_WTP_JEE_016/search.jhtml
你输入的关键字是: ??????
EncodingFilter获得请求: /PRJ_WTP_JEE_016/search.jhtml
你输入的关键字是: 上海

图 3-1-2

1-2. 任务目标：

1-2.1. 确保所有客户端发送的POST请求数据中，中文字符无乱码。

1-3. 任务要求：

1-3.1. 说明1. POST类型请求的特点是，请求数据通过请求体发送到服务端。

1-3.2. 说明2. 导致中文数据乱码的原因是：Tomcat默认请求体编码集不是UTF-8。

1-3.3. 通过EncodingFilter类，实现POST请求的编码集始终为：UTF-8：

1) 限制1. 编码集UTF-8需保存于web.xml文件中。

2) 限制2. 编码集需由EncodingFilter读取后，再行设置。

2. 实现思路：

1-1. 将编码集字符：utf-8 保存于 web.xml 中：

1-1.1. 在 web.xml 文件中，创建 EncodingFilter 的 init-param 节点。

1) param-name 节点建议设置为：encoding

2) param-value 节点设置为：utf-8

1-2. EncodingFilter 读取编码集。

1-2.1. 在 EncodingFilter 的 **init** 方法中，通过 **FilterConfig** 读取编码集。

1-3. EncodingFilter 设置编码集。

1-3.1. 在 EncodingFilter 的 **doFilter** 方法中，设置 Request 的编码集为 utf-8。

3. 推荐步骤：

3-1. 未编码前，观察乱码效果：

3-1.1. 运行项目工程。

3-1.2. 在浏览器地址栏输入 http://localhost/PRJ_WTP_JEE_016/ 访问首页。

3-1.3. 在页面顶部的**关键字搜索框**中输入：上海，点击搜索。

3-1.4. 观察控制台，见：你输入的关键字是：?????

3-1.5. 同时产品搜索页，没有显示任何数据（关键字乱码导致）。

3-2. 按实现思路，依次实现处理 POST 类型请求中，中文字符乱码的问题。

4. 验证与测试：

4-1. 运行项目工程。

4-2. 在浏览器地址栏输入 http://localhost/PRJ_WTP_JEE_016/ 访问首页。

4-3. 在页面顶部的关键字搜索框中输入：上海，点击搜索。

4-4. 观察控制台，见：你输入的关键字是：上海

4-5. 同时产品搜索页，显示所有与上海有关的旅游产品记录。

任务 3、处理 GET 类型请求中的乱码

1. 任务说明：

1-1. 完成效果：

```
EncodingFilter获得请求: /PRJ_WTP_JEE_016/image_cache/39.jpg
EncodingFilter获得请求: /PRJ_WTP_JEE_016/image_cache/45.jpg
EncodingFilter获得请求: /PRJ_WTP_JEE_016/search.jhtml
你输入的关键字是: ??????·
EncodingFilter获得请求: /PRJ_WTP_JEE_016/search.jhtml
你输入的关键字是: 上海
```

图 3-1-2

1-2. 任务目标：

1-2.1. 确保所有客户端发送的GET请求数据中，中文字符无乱码。

1-3. 任务要求：

1-3.1. 说明1. GET类型请求的特点是，请求数据通过URL发送到服务端。

1-3.2. 说明2. 由于GET类型请求的数据不通过请求体发送，因此任务2的业务操作对GET类型请求无效。

1-3.3. 通过server.xml文件的设置，实现GET请求的编码集始终为：UTF-8。

2. 实现思路：

2-1. 通过 server.xml 文件的设置，实现 GET 请求的编码集始终为：UTF-8：

2-1.1. 定位：Eclipse 开发环境中的 Server 工程。

2-1.2. 修改该工程中的 server.xml 文件（Tomcat 的核心配置文件）。

2-1.3. 定位：<connector port="80">节点。

2-1.4. 为其增加 URLEncoder 属性，属性值为：utf-8。

3. 推荐步骤：

3-1. 未编码前，观察乱码效果：

3-1.1. 运行项目工程。

3-1.2. 输入：http://localhost/PRJ_WTP_JEE_016/search.jhtml?key=上海

3-1.3. 观察控制台，见：[你输入的关键字是：?????](#)

3-1.4. 同时产品搜索页，没有显示任何数据（关键字乱码导致）。

3-2. 按实现思路，依次实现处理 GET 类型请求中，中文字符乱码的问题。

4. 验证与测试：

4-1. 运行项目工程。

4-2. 输入：http://localhost/PRJ_WTP_JEE_016/search.jhtml?key=上海

4-3. 观察控制台，见：[你输入的关键字是：上海](#)

4-4. 同时产品搜索页，显示所有与上海有关的旅游产品记录。

作者：Roger.Huang