Atitit gzip zip压缩过滤器 的模块功能语实现attilax总结

[1.1. 一般可以使用webserver自带的实现，。如果实现的不好或者不好配置的，或者需要精细化控制的，可以自己使用过滤器实现。。 1](#_Toc20133)

[1.2. 常见的压缩选项 1](#_Toc16744)

[1.3. 代码 1](#_Toc18235)

## 一般可以使用webserver自带的实现，。如果实现的不好或者不好配置的，或者需要精细化控制的，可以自己使用过滤器实现。。

我哦是因为tmcat插件模式不生效，只好自己实现filtrer

## 常见的压缩选项

compression="on"

compressionMinSize="10" 排除小文件

noCompressionUserAgents="traviata1 atibrow11"

compressableMimeType

文件扩展名过滤

文件路径过滤 排除 指定路径下的压缩

压缩模式 gzip **Accept-Encoding:**

gzip, deflate, br

## 代码

package com.cnhis.cloudhealth.clidoctorweb.advicevaluation;

import java.io.ByteArrayOutputStream;

//package me.gacl.web.filter;

import java.io.IOException;

import java.util.zip.GZIPOutputStream;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* 使用注解标注过滤器

\*

\* @WebFilter将一个实现了javax.servlet.Filte接口的类定义为过滤器 属性filterName声明过滤器的名称,可选

\* 属性urlPatterns指定要过滤

\* 的URL模式,也可使用属性value来声明.(

\* 指定要过滤的URL模式是必选属性)

\*/

@WebFilter(filterName = "zipFilter1", urlPatterns = "/\*")

public class zipFilter1 implements Filter {

@Override

public void destroy() {

System.out.println("过滤器销毁");

}

@Override

public void doFilter(ServletRequest request, ServletResponse response1, FilterChain chain)

throws IOException, ServletException {

System.out.println("---request.getServletContext():"+request.getServletContext());

System.out.println("---执行过滤操作");

HttpServletResponse response = (HttpServletResponse) response1;

// 创建HttpServletResponse 包装类的实例

MyHttpServletResponse myResponse = new MyHttpServletResponse((HttpServletResponse) response);

chain.doFilter(request, myResponse);

// GZIP压缩：

byte[] buff = myResponse.getBufferedBytes();

// 创建缓存容器：

ByteArrayOutputStream baos = new ByteArrayOutputStream();

GZIPOutputStream gzip = new GZIPOutputStream(baos);

gzip.write(buff);

gzip.close();

byte[] buff\_ziped = baos.toByteArray();

// 设置响应头；

response.setHeader("Content-Encoding", "gzip");

response.setContentLength(buff\_ziped.length);

response.getOutputStream().write(buff\_ziped);

}

@Override

public void init(FilterConfig config) throws ServletException {

System.out.println("--过滤器初始化");

}

}