Atitit.url 汉字中文路径 404 resin4 resin 解决 v2 q329

[1. Pluginx机制 1](#_Toc22238)

[2. Code 1](#_Toc29284)

[3. 参考 4](#_Toc1777)

# 原理

过滤器 ,,拦截jpg with %的url...read img ,直接输出..

# Pluginx机制

//if()

**if**(*inied*==**false** || **new** File("c:\\chinese\_filename\_hook\_ini\_pertime").exists()) {

**new** Pluginx().remove\_all\_filters("chinese\_filename\_hook");

**new** Pluginx().add\_action("chinese\_filename\_hook", **new** JpgFilerProcesser());

**new** Pluginx().add\_action("chinese\_filename\_hook", **new** TxtFilerProcesser());

*inied*=**true**;

}

}

作者::  ★(attilax)>>>  绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# Code

**package** com.attilax.corePkg;

**import** java.awt.image.BufferedImage;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.io.UnsupportedEncodingException;

**import** java.net.URLDecoder;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.imageio.IIOException;

**import** javax.imageio.ImageIO;

**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** aaaCfg.IocX;

**import** com.attilax.core;

**import** com.attilax.jsonX;

**import** com.attilax.io.filex;

**import** com.attilax.io.pathx;

**import** com.attilax.lang.Global;

**import** com.attilax.lang.Pluginx;

**import** com.attilax.net.JpgFilerProcesser;

**import** com.google.common.collect.Maps;

**import** com.sun.image.codec.jpeg.JPEGCodec;

/\*\*

\* com.focusx.util

\* EncodingFilter.java

\* author:vincente 2013-8-19

\*/

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

**public** **class** **ChineseCharFileUrlFilter** **implements** Filter {

**private** **String** charset;

**public** **void** **destroy**() {

// **TODO** Auto-generated method stub

}

***@SuppressWarnings***("all")

**public** **void** **doFilter**(ServletRequest request, ServletResponse response,

FilterChain chain) **throws** **IOException**, **ServletException** {

**try** {

**if**(**Global**.*map*.get()==**null**)

**Global**.*map*.set(**new** HashMap());

**Global**.*map*.get().put("FileSeverDir", "Z:\\动作类");

**new** Pluginx().remove\_all\_filters("chinese\_filename\_hook");

**new** Pluginx().add\_action("chinese\_filename\_hook", **new** JpgFilerProcesser(),0,**null**,**null**);

// new Pluginx().add\_action("chinese\_filename\_hook", new TxtFilerProcesser());

Map **m**=**new** HashMap ();

m.put("req", request);

m.put("res", response);

**new** Pluginx().do\_action("chinese\_filename\_hook",m,**null**);

} **catch** (**Exception** **e**) {

e.printStackTrace();

}

**core**.*log*("--loadorderO9::EncodingFilter");

//用init方法取得的charset覆盖被拦截下来的request对象的charset

request.setCharacterEncoding(**this**.charset);

//将请求移交给下一下过滤器，如果还有的情况下。

chain.doFilter(request, response);

}

// private String getImgPath(String url2) {

// String str;

// try {

// str = new String(url2.toString().getBytes("iso8859\_1"), "GB2312");

// } catch (UnsupportedEncodingException e) {

// e.printStackTrace();

// throw new RuntimeException(e);

// }

// String f = pathx.webAppPath() + File.separator + str;

// File file = new File(f);

// return f;

// }

**public** **void** **init**(FilterConfig config) **throws** **ServletException** {

//从web.xml中的filter的配制信息中取得字符集

**this**.charset = config.getInitParameter("charset");

}

}

# 参考

url汉字中文路径404异常resin4 resin chinese char path 404 err解决