

```

package path.to.filters;

import java.io.IOException;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

/**
 *
 * Listen for every call and filter every jsp request redirecting to a forbidden
page
 * if not in public (or deeper) directory.
 */
public class JspFilter implements Filter {
    private static final Log logger = LogFactory.getLog(JspFilter.class);
    private FilterConfig config = null;
    /* (non-Javadoc)
     * @see javax.servlet.Filter#init(javax.servlet.FilterConfig)
     */
    public void init(FilterConfig config) throws ServletException {
        this.config = config;
    }

    /* (non-Javadoc)
     * @see javax.servlet.Filter#doFilter(javax.servlet.ServletRequest,
javax.servlet.ServletResponse, javax.servlet.FilterChain)
     */
    public void doFilter(ServletRequest request, ServletResponse
response, FilterChain chain) throws IOException, ServletException {
        logger.debug("JspFilter::doFilter()");

        HttpServletRequest http = (HttpServletRequest) request;

        //if in "/public/" directory normal chain
        if(http.getRequestURI().indexOf("/public/")>=0){
            chain.doFilter(request, response);
        }else{
            RequestDispatcher disp =
http.getRequestDispatcher(config.getInitParameter("forbidUrl"));
            disp.forward(request, response);
        }
    }

    /* (non-Javadoc)
     * @see javax.servlet.Filter#destroy()
     */
    public void destroy() {
    }
}

```