

# ShortcutUrls

NOTE: This is outdated information that applies only to Tapestry 4, though the approach *may* be adaptable to Tapestry 5 as well.

This is a code for a Servlet Filter that allows to create shortcuts to Tapestry urls. It is an alternative to urlrewrite. I wrote this code because I'm coding most parameters using the datasqueezer and my shortcuts needs to do more processing than urlrewrite (also I'm allergic to regular expressions).

Inside the web.xml I have

```
<filter>
  <filter-name>ActServletFilter</filter-name>
  <filter-class>actualis.web.ServletFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>ActServletFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
public class ServletFilter implements Filter {

    private FilterConfig m_config;

    public void init(FilterConfig config) throws ServletException {
        m_config = config;
        for (UrlPage ipage:UrlPage.values()) {
            ipage.register(this);
        }
    }

    private DataSqueezer m_datasqueezer;

    public DataSqueezer getDataSqueezer() {
        if (m_datasqueezer == null)
            m_datasqueezer = new DataSqueezer(null);
        return m_datasqueezer;
    }

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException,
        ServletException {
        final HttpServletRequest hrequest = (HttpServletRequest) request;
        final HttpServletResponse hresponse = (HttpServletResponse) response;

        String servletPath = hrequest.getServletPath();

        String compString = servletPath.toUpperCase();
        for (UrlPage ipage:UrlPage.values()) {
            String cString = '/' + ipage.getName();
            if (compString.startsWith(cString.toUpperCase())) {
                ExternalPageInfos infos = ipage.getPageInfos(hrequest);
                String newURL = hrequest.getContextPath() + "/exec?service=external/" + infos.pageName;
                for (String par:infos.parameters) {
                    newURL = newURL + "&" + "sp=" + par;
                }

                hresponse.sendRedirect(newURL);
                return;
            }
        }

        chain.doFilter(request, response);
    }

    public void destroy() {
    }
}
```

And here is my enum that contains all the shortcut infos. It creates shortcuts to two Tapestry pages, "FicheArticle" and "Partenaires". The first one is a shortcut `article?ref=xxx` and calls "FicheArticle". It passes the ref and a 0 as integer to the external page. The second is a shortcut `partenaire?part=xxx` and calls a page "Partenaires". The `getSqueezeInfo` method creates a hibernate object reference for the `datasqueezer` that can then be loaded (using the code I posted on the [Hibernate/Spring/Tapestry integration](#)).

```
import java.io.Serializable;
import java.io.IOException;
import javax.servlet.http.HttpServletRequest;

import org.hibernate.Hibernate;

public enum UrlPage {

    Article("article") {
        public ExternalPageInfos getPageInfos(HttpServletRequest request) throws IOException {
            ExternalPageInfos infos = new ExternalPageInfos();
            infos.pageName = "FicheArticle";
            String idArt = getSqueezeInfo(Article.class, request.getParameter("ref"));
            infos.parameters = new String[] { idArt, squeeze(0) };
            return infos;
        }
    },
    Partenaires("partenaires") {
        public ExternalPageInfos getPageInfos(HttpServletRequest request) {
            ExternalPageInfos infos = new ExternalPageInfos();
            infos.pageName = "Partenaires";
            String id_part = request.getParameter("part");
            infos.parameters = new String[] { id_part };
            return infos;
        }
    };

    private String m_urlname;

    private static ServletFilter s_filter;

    private static String squeeze(Object obj) throws IOException {
        return s_filter.getDataSqueezer().squeeze(obj);
    }

    private static String getSqueezeInfo(Class clazz, Object id) throws IOException {
        String info = clazz.getName();
        info = info.concat(ActualisEngine.HIB_SEPARATOR);
        info = info.concat(s_filter.getDataSqueezer().squeeze(id));
        return ActualisEngine.HIB_PREFIX + info;
    }

    UrlPage(String name) {
        m_urlname = name;
    }

    public String getName() {
        return m_urlname;
    }

    public abstract ExternalPageInfos getPageInfos(HttpServletRequest request)
        throws IOException;

    public void register(ServletFilter filter) {
        s_filter = filter;
    }
}
```

And the last class

```
public class ExternalPageInfos {  
    String pageName;  
    String[] parameters;  
}
```