

EncodeURL Transformer (2.1 legacy document)

Table of contents

1 Comments.....4

Table of contents

1 EncodeURL Transformer.....	3
------------------------------	---

Warning:

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. EncodeURL Transformer

The encodeURL transformer emits encoded URLs. This transformer applies encodeURL method to URLs. You may want to use this transform to avoid doing the manually encodeURL() call.

Usually this transformer is appended as last transformer before the serialization process. In this case it is possible to encode URLs introduced in the generator, and xslt transformer phase.

You can specify which attributes hold URL values in order to restrict URL rewriting to specific attributes and values only. In the current implementation you specify include, and exclude patterns as regular expressions, concatenating element-name-regexp + "/" + attribute-name-regexp [+ "=" + attribute-value-regexp] whereas the part in brackets [] are optional.

The EncodeURLTransformer has several configuration options. These options may be specified in the sitemap, or by each request.

include-name

RE pattern for including attributes for encode URL rewriting, The attribute values are encoded, if an expressions of the form

element-name-regexp/@attribute-name-regexp[=attribute-value-regexp] matches.

By default include-name is defined as .*/@href|.*/@action|frame/@src.

exclude-name

RE pattern for excluding attributes from encode URL rewriting, The attribute values are not encoded, if an expressions of the form

element-name-regexp/@attribute-name-regexp[=attribute-value-regexp] matches.

By default exclude-name is defined as img/@src.

- Name : encodeURL
- Class: org.apache.cocoon.transformation.EncodeURLTransformer
- Cacheable: yes.

A simple example might help to use the EncodeURLTransformer effectively:

Add the EncodeURLTransformer to the components in your sitemap.xmap

```
... <map:components> ... <map:transformers default="xslt"> ... <map:transformer
name="encodeURL" src="org.apache.cocoon.transformation.EncodeURLTransformer"> <!--
default configuration, explicitly defined -->
<include-name>.*\/@href|.*/@action|frame/@src</include-name>
<exclude-name>img\/@src|(a\/@href|iframe\/@src)=.*adserver</exclude-name>
</map:transformer> ...
```

Next define in your pipeline to use the EncodeURLTransformer

```
<map:match pattern="*.xsp"> <map:generate type="serverpages"
name="docs/samples/xsp/{1}.xsp"/> <map:transform
src="stylesheets/page/simple-page2html.xml"/> <map:transform type="encodeURL"/>
<map:serialize/> </map:match>
```

In this example pipeline it is assumed that the attribute href of element a contains an URL which should get encoded. Moreover the attribute action of any element contains an URL which should get encoded, too. Finally the attribute src of element frame should get encoded, too.

The attribute `src` of element `img` is excluded from encoding.

In other words, images are served regardless of the current session, in contrast anchor links, form actions, and frame `src` are served depending on the current session.

The encoding itself applies the servlet method `response.encodeURL()` upon the URL.

1. Comments

add your comments