

HtmlCrossReferenceAuditor

 Added in v5.6.6

The `HtmlCrossReferenceAuditor` compares HTML elements within the same page. You can use `HtmlCrossReferenceAuditor` to compare the meta keywords to page title or meta description.

The auditor uses `jsoup` queries to parse and find HTML elements. These queries have a jQuery or CSS like syntax.

`HtmlCrossReferenceAuditor` uses two `jsoup` queries:

- A *source* query to retrieve some text
- A reference query to retrieve one or HTML elements to be checked against the source text

The source query also has a regular expression to extract the source text. The reference query also has a regular expression that checks the matched references; if the references match the regular expression, the audit is passed.

Here's a quick example:

source query (`sourceQuery`): `meta[name="keywords"]`

A `jsoup` query to find the meta keywords element in a page.

source pattern (`sourcePattern`): `.+content="([^,]+), .+`

A regular expression to search the meta keywords element found by the source query and match the first keyword as match group 1.

The source query and the source pattern will get the first keyword defined in the meta keywords element.

reference query (`referenceQuery`): `meta[name="description"]`

A `jsoup` query will find the meta description in a page.

reference pattern (`referencePattern`): `<meta name="description" content=".'{0,}'{0}.'{0,}'">`

A regular expression to check that the keyword found by the source query and pattern appears in the content attribute of the meta description.

Suppose the meta keywords element of the page is:

```
<meta name="keywords" content="beach,resort,island" />
```

The result of the source query and pattern is "beach". The result "beach" will be replaced in the reference pattern and applied to meta description content. If "beach" is found, the audit is passed, if not, the audit fails.

Class: `info.magnolia.services.seo.audit.impl.HtmlCrossReferenceAuditor`

In addition to the common auditor properties, this auditor can be configured with the following properties:

level	required Determines how a failed audit will be counted: <ul style="list-style-type: none">• Error (<code>auditErrors</code>)• Warning (<code>auditWarnings</code>)• Note (<code>auditNotes</code>)
auditProperty	required Defines the property name for storing failed audit results.  The property name should be unique among auditors or auditors may overwrite results.

auditValue	<p>required</p> <p>Defines a message or explanation for a failed audit.</p> <p>The message can have placeholders that are replaced with information about the node and auditor:</p> <ul style="list-style-type: none"> • 0 - the source text found by <code>sourceQuery</code> and <code>sourcePattern</code> • 1 - the reference text found by <code>referenceQuery</code>
sourceQuery	<p>required</p> <p>A valid jsoup query. See this cookbook for more on jsoup queries.</p>
sourceText	<p>optional</p> <p>Controls whether the source pattern will be applied to the HTML element found by <code>sourceQuery</code> (when set to <code>false</code>) or the text of the HTML element (when set to <code>true</code>).</p>
sourcePattern	<p>required</p> <p>A valid Java regular expression.</p> <p>You can use match groups in the regular expression.</p>
sourceFlags	<p>optional</p> <p>Match flags, as defined by <code>java.util.regex.Pattern</code>.</p> <p>The value must be a bit mask that may include <code>Pattern.CASE_INSENSITIVE</code>, <code>Pattern.MULTILINE</code>, <code>Pattern.DOTALL</code>, <code>Pattern.UNICODE_CASE</code>, <code>Pattern.CANON_EQ</code>, <code>Pattern.UNIX_LINES</code>, <code>Pattern.LITERAL</code>, <code>Pattern.UNICODE_CHARACTER_CLASS</code> and <code>Pattern.COMMENTS</code>.</p>
sourceGroup	<p>optional</p> <p>The index of the match group to use as source text.</p> <p>If not specified or set to 0, the entire matching text of <code>sourcePattern</code> will be used.</p>
referenceQuery	<p>required</p> <p>A valid jsoup query. See this cookbook for more on jsoup queries.</p> <p>The result of the reference query will be compared to the source query by the <code>referencePattern</code>.</p>
referenceText	<p>optional</p> <p>Controls whether the reference pattern will be applied to the HTML element found by <code>referenceQuery</code> (when set to <code>false</code>) or the text of the HTML element (when set to <code>true</code>).</p>
referencePattern	<p>required</p> <p>A valid Java regular expression.</p> <p>The <code>referencePattern</code> can have placeholders that are replaced by the source text found by <code>sourceQuery</code> and <code>sourcePattern</code>:</p> <ul style="list-style-type: none"> ▪ 0 - the source text <p>⚠ IMPORTANT: the source text is substituted into the reference pattern before the reference pattern is applied to the reference text. You should wrap brackets in your <code>referencePattern</code> regular expression in single quotes to prevent them being interpreted as a placeholder.</p>
referenceFlags	<p>optional</p> <p>The value must be a bit mask that may include <code>Pattern.CASE_INSENSITIVE</code>, <code>Pattern.MULTILINE</code>, <code>Pattern.DOTALL</code>, <code>Pattern.UNICODE_CASE</code>, <code>Pattern.CANON_EQ</code>, <code>Pattern.UNIX_LINES</code>, <code>Pattern.LITERAL</code>, <code>Pattern.UNICODE_CHARACTER_CLASS</code> and <code>Pattern.COMMENTS</code>.</p>
fetcher	<p>required</p> <p>Defines the content fetcher for the selected node. The query is then applied to the fetched content.</p> <p>There are two types of content fetchers available, for more about content fetchers, see: Content Fetchers.</p>

Example

Here is an example from the SEO module. You can find this configuration here: `/modules/seo/config/auditManager/auditors/titleRendered`

```
checkMetadescriptionKeyword:
  auditProperty: checkMetadescriptionKeyword
  auditValue: The main keyword {0} is not used in the page metadescription
  class: info.magnolia.services.seo.audit.impl.HtmlCrossReferenceAuditor
  description: Check if the main keyword is used in the metadescription (pre-prod)
  level: auditWarnings
  referenceFlags: 2
  referencePattern: <meta name="description" content=".'{0,}'{0}.'{0,}'">
  referenceQuery: meta[name="description"]
  sourceGroup: 1
  sourcePattern: .+content="([\^,]+),.+
  sourceQuery: meta[name="keywords"]
  fetcher:
    class: info.magnolia.services.seo.audit.impl.RequestFetcher
  targets:
    localhost:
      class: info.magnolia.services.seo.audit.impl.HostTarget
      host: localhost
      password: superuser
      port: 8080
      scheme: http
      user: superuser
```