

# Content Fetchers

 Added in v5.6.2

Some auditors check on rendered content rather than the content node itself. These auditors need some way to get the content before checking it and use a "content fetcher" to retrieve the content for them.

These auditors use content fetchers:

- [HtmlElementAuditor](#) - to check rendered HTML elements
- [LinkAuditor](#) - to check rendered links
- [ParagraphLengthAuditor](#) - to check rendered text
- [ValidHtmlAuditor](#) - to check the HTML of a rendered page

Each of these auditors must have a content fetcher configured to carry out its checks.

- [RequestFetcher](#)
  - [Configuring credentials](#)
  - [Example](#)
- [RenderingFetcher](#)
  - [Example](#)

## RequestFetcher

`RequestFetcher` sends a preview request to Magnolia to render content and captures the page returned. You have to configure `RequestFetcher` to send a preview request through its configuration properties.

**Class:** `info.magnolia.services.seo.audit.impl.RequestFetcher`

`RequestFetcher` has the following properties:

<code>requestTemplate</code>	<p><b>optional</b></p> <p>The template for the preview request. The template should have a placeholder "{0}" for the path of the selected content node being rendered.</p> <p>The default value of <code>requestTemplate</code> is:</p> <pre>http://localhost:8080/{0}?mgnlPreview=true&amp;mgnlChannel=desktop</pre> <p>Note that the default <code>requestTemplate</code> will send its preview request to the local Magnolia instance listening on port 8080.</p> <p>Also, the query string of the request template specifies that it is a preview request ("<code>mgnlPreview=true</code>") and to render the page as if being viewed in a desktop browser ("<code>mgnlChannel=desktop</code>").</p> <p>You could also use the personalization preview parameters in the query string of the request template to preview the page as a given persona:</p> <pre>http://localhost:8080/{0}?previewPersona=&lt;persona name&gt;&amp;mgnlPreviewAsVisitor=true</pre>
<code>targets</code>	<p><b>optional</b></p> <p>Defines a list of credentials (host, port, user name and password) when requesting rendered content. The credentials will be added to the preview request and will be available as preemptive basic authentication.</p> <p>Note that accessing pages on the standard Magnolia author instance require authentication. You should configure credentials for accessing your local Magnolia author instance.</p>

## Configuring credentials

The `targets` nodes allows you to define one or more credentials that will be added to each preview request used to render content. You will probably have to define credentials for accessing links to your Magnolia instance if it is an author instance; pages and resources are protected with basic authentication.

The credentials will be added by host, so more than one credentials can be added.

Here is how to configure target credentials within your `RequestFetcher` configuration:

targets	
<target credentials name>	<b>required</b> The name of the credentials
class	<b>required</b> The credentials class name, should be <code>info.magnolia.services.seo.audit.impl.HostTarget</code>
host	<b>required</b> The host or domain name for the credentials
port	<b>required</b> The port
scheme	<b>optional</b> The scheme (e.g. <code>http</code> , <code>https</code> , ...) The default value for <code>scheme</code> is <code>http</code> .
user	<b>required</b> The user name
password	<b>required</b> The user's password
preemptive	<b>optional</b> If <code>true</code> the credentials will be added to requests with preemptive basic authentication. The default value for <code>preemptive</code> is <code>true</code> .

## Example

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

```
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
```

## RenderingFetcher

**Class:** `info.magnolia.services.seo.audit.impl.RenderingFetcher`

`RenderingFetcher` renders content by directly invoking the Magnolia rendering engine and capturing the results.

Since `RenderingFetcher` does not send a request to Magnolia. It may be slightly faster than `RequestFetcher` but may not correctly render the content when using personalization or sites.

`RenderingFetcher` has no configuration properties.

## Example

Here is an example RenderingFetcher configuration:

```
fetcher:  
  class: info.magnolia.services.seo.audit.impl.RenderingFetcher
```