Search Engine Optimization

The Search Engine Optimization (SEO) module provides tools for analyzing and validating content in your JCR repository. Typically it is used to analyze pages but it could also be used to analyze any type of content node in any workspace. The module contains a variety of audit routines that can be configured to test different aspects of your pages and/or content. It has an app which allows content managers to run audit routines and view the results. These routines can be executed independently or as part of package using the Audit Manager.

- Installation
- Usage
    - SEO Tuner app
        - Browser subapp
        - Detail subapp
    - Actions
    - Auditors
    - Content Fetchers
- Warnings
- Changelog

Status	**INCUBATOR**
Type	Visual App
Edition	EE Pro
JIRA	SEO
Git	search-engine-optimization

Installation

Maven is the easiest way to install the module. Add the following dependency to your bundle:

```
<dependency>
  <groupId>info.magnolia.seo</groupId>
  <artifactId>magnolia-seo</artifactId>
  <version>${seoVersion}</version>
</dependency>
```

This module is located in the incubator repository and requires additional changes to your Maven settings file.

Versions

5.6.6	Magnolia 5.6
5.6.5	Magnolia 5.6
5.6.4	Magnolia 5.6
5.6.3	Magnolia 5.6

Usage

The module contains different audit routines that can be run to validate content. Each routine is referred to as an **auditor**.

An auditor is a Java class that can:

- Test the rendered HTML of your page for validity.
- Test if required properties are defined for your page or content according to best practice SEO.
- Connect to an external service to analyze your page.
- be extended and customized to any analysis you need

Auditors analyze a selected node and either pass or fail the node. Audit failures are graded at three different levels:

- **Error** - The problem must be corrected.
- **Warning** - The problem should be corrected.
- **Note** - The problem is not severe but could be corrected.

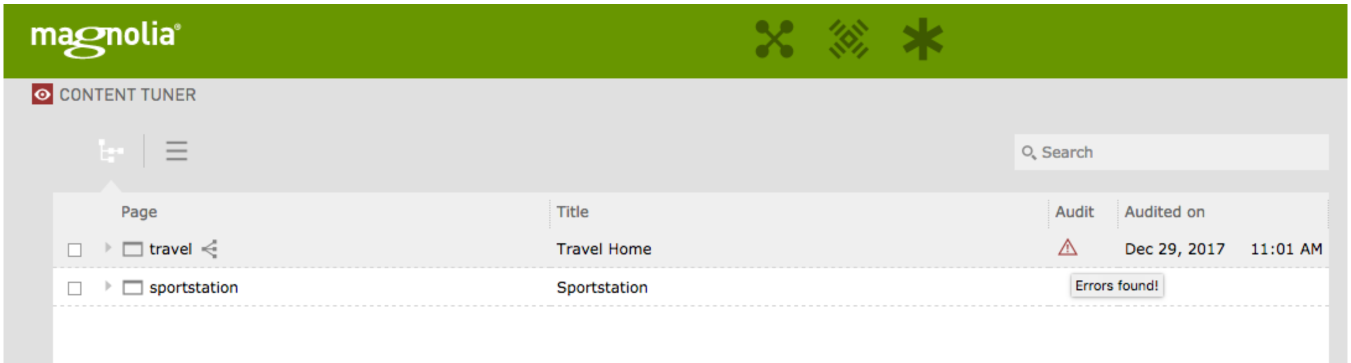
The ability to run audits is handled by the Audit Manager. It provides a framework for executing one or more auditors and saving their results. You can add new auditors and change the configuration of defined auditors from the Audit Manager.

SEO Tuner app

The SEO Tuner app (Audit app) is where users can launch audits on web pages. You will find the app installed in the Edit group of admincentral. From the app you can view the results of audits as well as export results to a file.

Browser subapp

The browser subapp shows all sites within the website workspace. It adds two new columns for audit status. If the page has errors, warnings or notes this will be indicated in the **Audit** column. The last time an audit was ran will be reflected in the **Audited on** column.

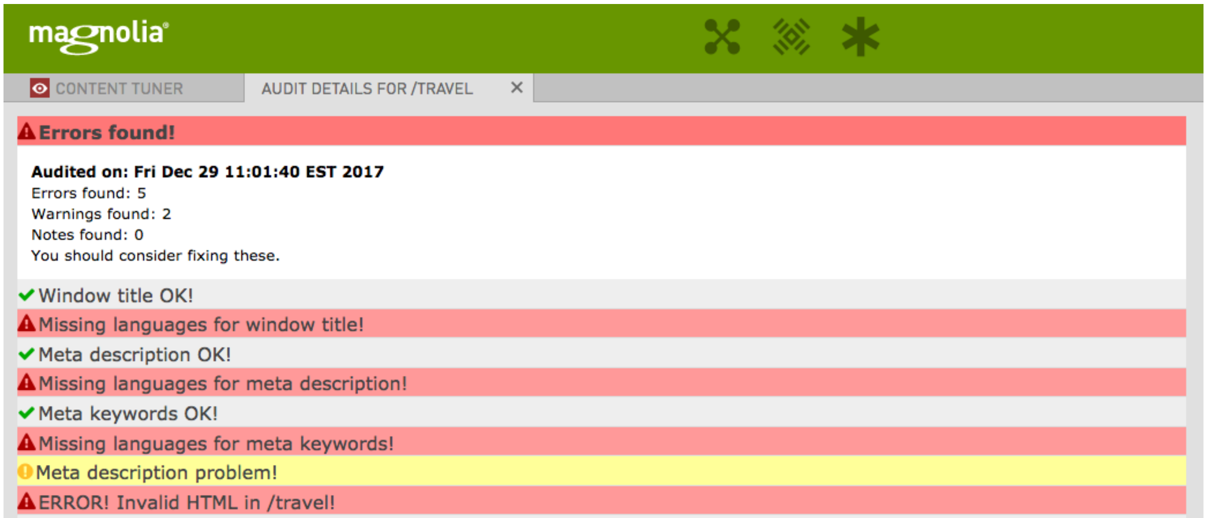


	Page	Title	Audit	Audited on
□	travel	Travel Home	⚠	Dec 29, 2017 11:01 AM
□	sportstation	Sportstation	Errors found!	

Detail subapp

This subapp gives information about the audit results. This subapp is spawned when an audit has been run on a selected page from the browser subapp. The audit detail view shows you an overview detailing how many errors, warnings and notes were found in the last audit. The audit details view also has a section for each successful or failed audit that was performed.

Click on an audit section to expand it and find more details on problem found. For some audits, you might also find a button that will link you to an app where you can correct the problem.



Errors found!

Audited on: Fri Dec 29 11:01:40 EST 2017
Errors found: 5
Warnings found: 2
Notes found: 0
You should consider fixing these.

✓ Window title OK!
⚠ Missing languages for window title!
✓ Meta description OK!
⚠ Missing languages for meta description!
✓ Meta keywords OK!
⚠ Missing languages for meta keywords!
! Meta description problem!
⚠ ERROR! Invalid HTML in /travel!

Actions

The SEO module contains two custom actions that can be configured into other apps. Both actions launch an analysis by the Audit Manager using configured auditors:

- `info.magnolia.services.seo.action.AuditPageActionDefinition`

  Here it is as a sample decoration file:

  ```
  subApps:
    browser:
      actions:
        auditPage:
          class: info.magnolia.services.seo.action.AuditPageActionDefinition
          icon: icon-info
          label: Audit
          availability:
            rules:
              IsNotDeletedRule:
                implementationClass: info.magnolia.ui.framework.availability.IsNotDeletedRule
      actionbar:
        sections:
          pageActions:
            groups:
              editingActions:
                items:
                  auditPage:
                    name: auditPage
  ```

  > ⓘ `AuditPageAction` launches an analysis running all active auditors (see the active property for auditor configuration below).

- `info.magnolia.services.seo.action.FlexiAuditPageActionDefinition`

  > ⓘ `FlexiAuditPageAction` opens a dialog showing all currently configured auditors, allowing the user to pick and choose which auditors are run.

Auditors

The SEO module contains several types of individual auditors. Auditors can be categorized into validation and definition auditors. They separate the work of determining if a property is defined from the validity of its content. Another dimension of categorization is language specifications for the site, single vs multi language.

- **Definition auditors** - check that the value(s) for a property have been defined. They do not validate the value(s) of the property.
- **Validation auditors** - validate the value(s) for a designated property if the property (or its language variant) have been defined.

	Definitions auditors	**Validation auditors**
Single Language	PropertyDefinitionAuditor	PropertyValidationAuditor
Multi-Language	I18NPropertyDefinitionAuditor	I18NPropertyValidationAuditor

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 render the content before checking it and use a "content fetcher" to do the rendering for them.

You have two content fetchers that can be used with auditors:

RenderingFetcher	Calls Magnolia's rendering engine directly to render the selected content node and captures the results. RenderingFetcher will not add traits or additional information to the rendering. *Use when you are using Personalization or Magnolia's Site module to ensure that the rendered content is correct.*

RequestFetcher	Sends a request to Magnolia to preview the selected content node and captures the returned page. Since RequestFetcher is actually sending a request to Magnolia to preview a page, the request will be processed as a normal HTTP requests, personalization traits and other traits will be determined as usual when previewing a page. *Use when not using personalization or sites to gain a little performance boost in rendering.*

Warnings

- This module is at INCUBATOR level.

Changelog

- Version 5.6.6
  - added HtmlCrossReferenceAuditor
  - clean of YAML definitions and internationalization
  - corrected NullPointerExceptions when viewing audit results
- Version 5.6.5
  - Extensions to info.magnolia.services.seo.audit.impl.HtmlElementAuditor for checking values with regular expressions
  - Fixes to info.magnolia.services.seo.audit.impl.external.MozAuditor to handle connection failures
  - Sample configuration for TextGears and Moz services
- Version 5.6.4
  - Corrected app definition errors
- Version 5.6.3
  - Code cleanup
- Version 5.6.2
  - SEO-14 - Getting issue details... STATUS
- Version 5.6.1
  - Workflow integration
- Version 5.6 - Initial release of the extensions version of the module.
  - SEO-13 - Getting issue details... STATUS
  - SEO-12 - Getting issue details... STATUS
  - SEO-11 - Getting issue details... STATUS
  - SEO-10 - Getting issue details... STATUS
  - SEO-9 - Getting issue details... STATUS