

Content Recommender

The Content Recommender module provides relevant content based on the user's journey in real time. It allows for assigning scores based on page visits or how the user interacts with your content, such as, clicking links or playing media.

- [Installation](#)
- [Usage](#)
 - [Understanding score models](#)
 - [Score Models app](#)
 - [Assigning score models](#)
 - [Integrating score models in your content apps](#)
 - [Integrating score models in your templates](#)
 - [JavaScript](#)
 - [Templating functions](#)
 - [Scoring components](#)
 - [Personalizing based on interest](#)
 - [Using centralized analytics](#)
- [Warnings](#)
- [Changelog](#)

Status	ADDON
Type	Visual App
Edition	EE Pro
JIRA	RECOMMEND
Git	recommender

Installation

Maven is the easiest way to install the module. Add the following dependency to your [bundle](#):

```
<dependency>  
  <groupId>info.magnolia.recommendation</groupId>  
  <artifactId>magnolia-recommender</artifactId>  
  <version>${recommenderVersion}</version>  
</dependency>
```

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

Versions

1.0.4	Magnolia 5.6
1.0.3	Magnolia 5.6
1.0.2	Magnolia 5.6
1.0.1	Magnolia 5.6

Usage

Scoring helps to identify interest in certain products, product groups, services, or topics. The identified interest scores help you to display personalized experiences.

Understanding score models

Score models identify a customer's interest when they browse your site. Their interest is captured in the form of interest points. For example, events like clicking links or page visits indicates interest on certain topics. Watching a video or interaction with the page may indicate even higher interest. These important events make up a dimension of the score model.

Not every interaction indicates the same amount of interest. Therefore you can adjust the number of value points to be assigned per specific event. Score assignment can be fine-tuned by first visit, recency, and frequency. For example, frequent and recent visits normally indicate a strong interest in a topic and should be weighted higher than points assigned for a first time visit.

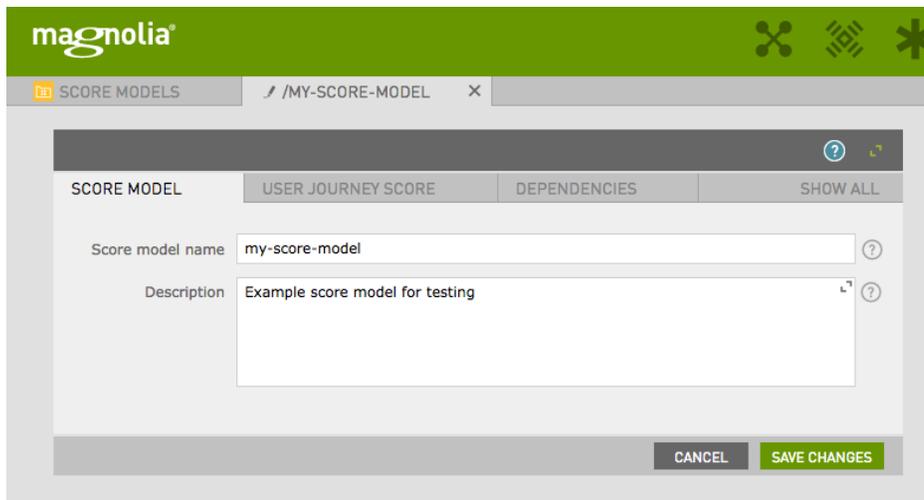
Together these events, and their weights, along with frequencies make up the score model.

Score Models app

The Score Models app is where you create and manage models. It's a good idea to arrange your models into folders. Perhaps Products, Campaigns, and /or Special offers.

When creating models pick a non technical name that is understandable for your editors and marketers. Additionally, you can give the model a description to explain what you intend to classify. The user journey score tab allows you to pre-configure your user score assignment. Therefore you don't have to configure the score on each and every content element where the model is applied.

The *dependencies* tab gives you an overview where your model is currently in use. This includes score models assigned for page views as well as component interactions.



Assigning score models

Score models can be assigned in a couple different ways. It can be applied directly in the Pages app or in your custom content apps.

- Page view scoring is handled in the browser subapp. Click *Edit Score Model*, then assign points for visiting that page.
- Interacting with components scoring is handled in the page editor. Use the *Score model assignment* tab in the component dialog.

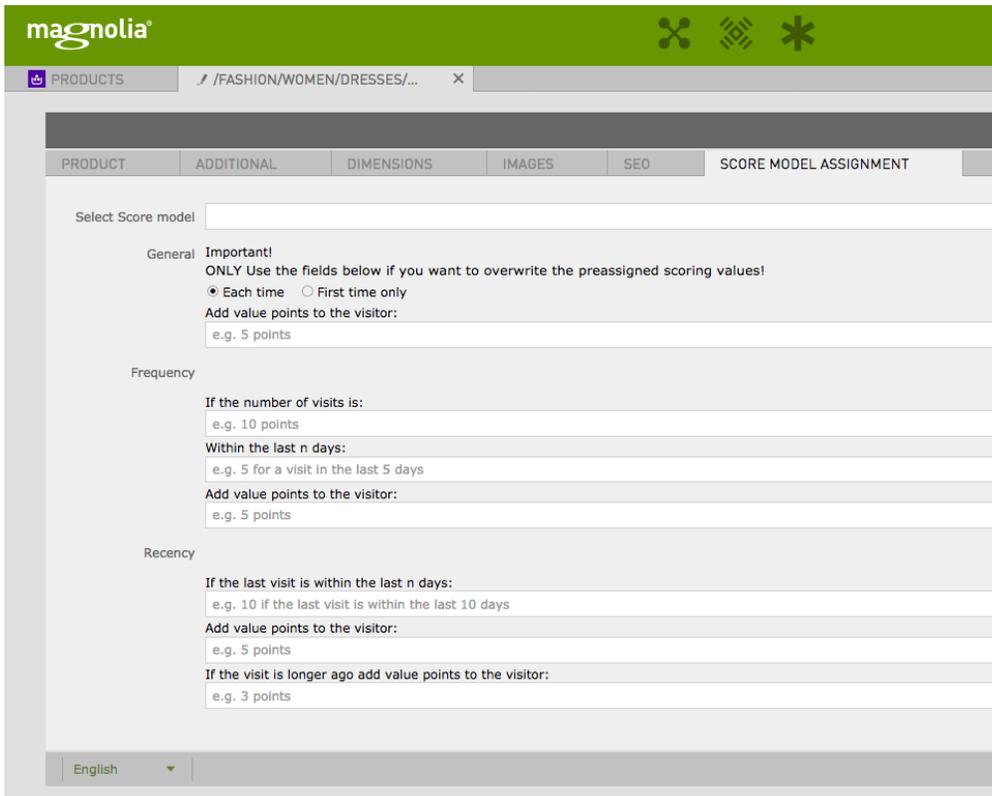
Integrating score models in your content apps

You can integrate scoring models in any content app. This allows you to tag your centrally managed content like products or teasers with scoring models. Each time a tagged content record is used the matching scoring tag is delivered as well. Therefore personalizing user journeys becomes very efficient.

To integrate the scoring model into your content app you can use a [decoration](#) to extend your apps:

```
/my-module/decorations/target-module/apps/app-name.yaml
```

```
subApps:
  detail:
    editor:
      form:
        tabs:
          - !include:/magnolia-recommender/dialogs/scoreModel.yaml
```



Integrating score models in your templates

The module provides requires some integration into you templating project.

JavaScript

The `recommender.js` script handles the persistence of the scoring on the browser side. Be sure to include this file provided by the content recommender module.

```
<script type="text/javascript" charset="utf-8" async="" src="/.resources/magnolia-recommender/webresources/js/recommender.js"></script>
```

Templating functions

The `recommender.js` script relies on HTML tags to understand which interactions should be tracked. The required tags can be generated with the help of the `RecommenderTemplatingFunctions`. They are available in the `scorefn` templating functions.

For example to enable page visit events, add the following to the HTML `<head>` section.

```
${scorefn.getPageVisitMetaTags(content)}
```

Scoring components

For specific components extend the component dialog. You can do so by including the `scoreModel.yaml` file.

```
form:
  tabs:
    - !include:/magnolia-recommender/dialogs/scoreModel.yaml
```

Next you need to generate the tags in your element by using the `scorefn` function.

```
<a class="yourButton" ${scorefn.getComponentScoreModelMetaTag(content)}>...
```

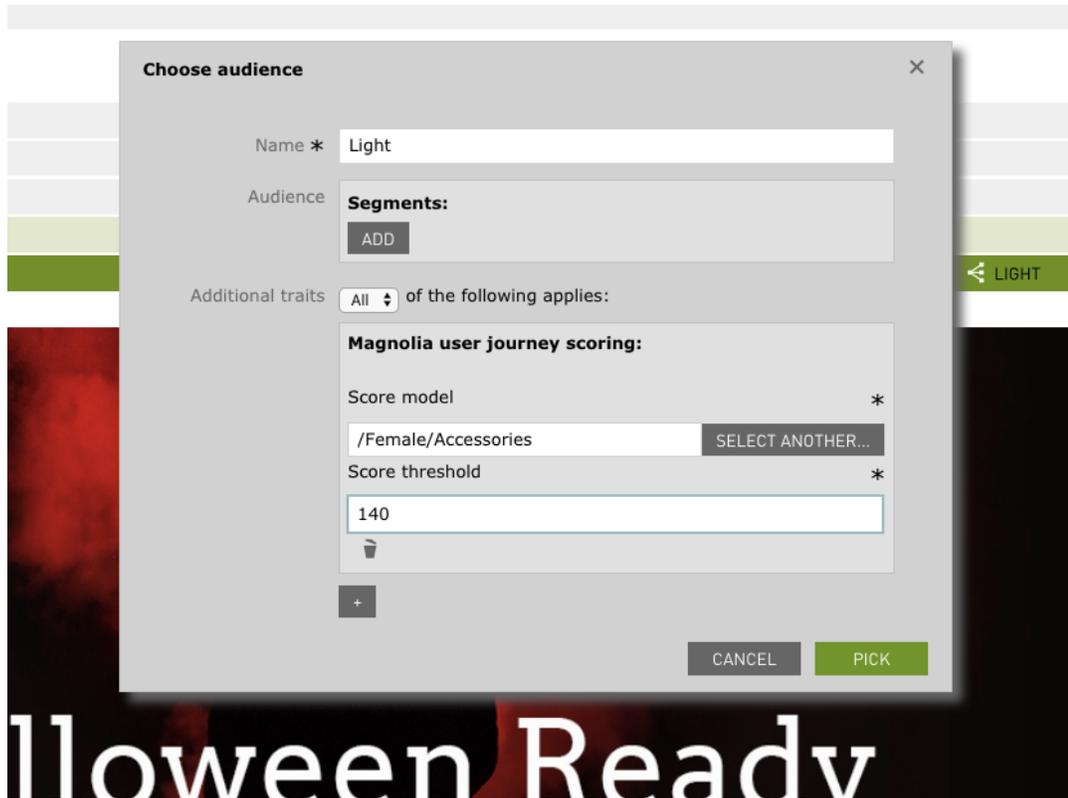
If you use content app data on your page then use the following code to generate the meta tags

```
<a class="yourButton" ${scorefn.getComponentScoreModelMetaTag("your node id", 'your node workspace')}}>...
```

Personalizing based on interest

To deliver personalized content create a component variant and assign a score model or a predefined segment containing score models.

If you assign a score model specify which interest threshold must be reached.



Using centralized analytics

The module is not collecting any analytics data. However, integration into your existing analytics solution is straight forward. Extend the `recommender.js` file and add custom tracking events in the `MgnlScoreModel#trackEvent()` method.

Here are some links that should help determine how to add the events.

- [Google analytics](#)
- [Piwick / Mamoto](#)
- [Omniture](#)

Warnings

- This module is at INCUBATOR level.

Changelog

- Version 1.0.5-SNAPSHOT

- [RECOMMEND-1 - Getting issue details...](#) STATUS

- Version 1.0.3
 - Skip this version.
- Version 1.0.4
 - Multi model assignment
- Version 1.0.2
 - Centralized score model assignment .
- Version 1.0.1 - Initial release of the extensions version of the module.