

CUE Semantic
User Guide
1.7.0-7

Table of Contents

- [1 Introduction.....](#) **3**
 - [1.1 How CUE Semantic Works.....](#) **3**
 - [1.2 Atilika Concepts and Entities.....](#) **3**
- [2 Installation.....](#) **5**
 - [2.1 Installation Procedure.....](#) **5**
 - [2.2 Configuration.....](#) **5**
 - [2.2.1 Configuring CUE Semantic.....](#) **6**
 - [2.2.2 Configuring CUE.....](#) **9**
 - [2.3 Running CUE Semantic.....](#) **11**
 - [2.4 Troubleshooting.....](#) **11**
- [3 Using CUE Semantic.....](#) **13**

1 Introduction

CUE Semantic is a CUE extension that provides semi-automatic content tagging. If CUE Semantic is installed, then a **Semantic Analysis** section is added to the content editor's metadata panel, and used to display a list of suggested tags for the content currently displayed in the editor.

The list of suggestions is constantly updated to reflect changes in the content. Users can then tag content items by simply selecting suggestions from this list.

CUE Semantic does not actually perform semantic analysis itself – rather, it offers seamless integration with an external semantic analysis service. In order to use CUE Semantic, therefore, you must separately purchase access to one of the semantic analysis services supported by CUE Semantic. At present the only service supported by CUE Semantic is [Atilika](#).

CUE Semantic is based on a Java framework for building web services called [Dropwizard](#).

1.1 How CUE Semantic Works

CUE Semantic works as follows:

- Content in the CUE editor is sent at regular intervals to the external service for analysis (every 10 seconds, by default).
- The external service returns a list of suggested tags for the submitted content, which are displayed in the **Semantic Analysis** section of the metadata panel.
- Information about tag selections made by users can be sent back to the external service. This information can then be used by the external service to improve future suggestions.

1.2 Atilika Concepts and Entities

The Atilika service returns two different types of tag suggestions:

Entities

Entities are tags referencing named entities in the real world, such as persons, places, organisations and so on. Atilika entities are defined by Atilika, and completely in Atilika's control. You cannot change the entity-based tag suggestions made by Atilika. Atilika entities are grouped into the following categories:

- **PER** (persons such as Nelson Mandela, Donald Trump, etc.)
- **LOC** (locations such as Africa, Spain, New York, Chelsea)
- **ORG** (organizations such as companies, charities, political parties and so on)
- **TTL** (titles such as Mr, Dr, Sir etc.)
- **FAC** (facilities such as hotels, sports grounds, theatres, universities, shopping malls)
- **REL** (religions)
- **NAT** (nationalities such as Irish, Indian, Australian)

Concepts

Concepts are more abstract tags, referencing ideas or topics such as politics, medicine, football, crime and so on. Concepts are not defined by Atilika, they are customer-defined tags. Atilika learns how to apply concepts by example. The process can be started by supplying a large number of ready-tagged documents to Atilika. Atilika learns from this sample what concepts to suggest for future content. If continuously supplied with feedback on how concept tags are used, it will continue learning and improving its suggestions.

CUE Semantic can be configured to make use of entities only, concepts only, or both entities and concepts.

2 Installation

You can install CUE Semantic on any Ubuntu or RedHat / CentOS machine in your network. It does not need to be installed on the same machine as CUE (although it can be). Before you try to install it, make sure that:

- You have installed Java 11 or higher on the machine where CUE Semantic is to be installed.
- You have credentials for accessing Escenic's software repositories.
- You have credentials for accessing the Atilika semantic analysis service.

2.1 Installation Procedure

To install CUE Semantic on an Ubuntu system, do the following:

1. Log in as **root**.
2. If necessary, add the Escenic **apt** repository to your list of sources:

```
# echo "deb http://user:password@apt.escenic.com stable main non-free" >> /etc/
apt/sources.list.d/escenic.list
```

where *user* and *password* are your Escenic download credentials (the same ones you use to access the Escenic Maven repository). If you do not have any download credentials, please contact [Escenic support](#).

3. Enter the following commands:
4. If you intend to use the feedback service to return training data to the semantic analysis service, then you also need to enter:

```
# apt-get update
# apt-get install cue-semantic-enrichment-service-1.7.0-7
```

```
# apt-get install cue-semantic-feedback-service-1.7.0-7
```

On RedHat / CentOS systems, enter the following command as **root**:

```
# rpm -Uvh https://user:password:yum.escenic.com/rpm/cue-semantic-enrichment-
service-1.7.0-7.x86_64.rpm
```

to install the main service, and if required:

```
# rpm -Uvh https://user:password:yum.escenic.com/rpm/cue-semantic-feedback-
service-1.7.0-7.x86_64.rpm
```

to install the feedback service

2.2 Configuration

After installation, you need to configure both CUE Semantic itself and CUE.

2.2.1 Configuring CUE Semantic

CUE Semantic consists of two components:

cue-semantic-enrichment-service

This service is responsible for getting tag suggestions from the external service and must be configured.

cue-semantic-feedback-service

This optional service is responsible for sending feedback to the external service. It only needs to be configured if you have installed it and intend to use it.

2.2.1.1 Configuring the Enrichment Service

To configure **cue-semantic-enrichment-service**:

1. Log in as **root**.
2. Open **/etc/escenic/cue-semantic-enrichment-service/cue-semantic-enrichment-service.yaml** in an editor.
3. Replace the **providers** group of settings with the following:

```
providers:
  - type: atilika_entities
    url: atilika-url
    relevanceThreshold: 0.3
    connectTimeout: 5000
    requestTimeout: 5000
    credentials:
      username: atilika-username
      password: atilika-password
  - type: atilika_concepts
    url: atilika-url
    relevanceThreshold: 0.3
    connectTimeout: 5000
    requestTimeout: 5000
    credentials:
      username: atilika-username
      password: atilika-password
```

where *atilika-url*, *atilika-username* and *atilika-password* are access credentials supplied by Atilika. The default settings shown above should work in most circumstances for the other properties:

relevanceThreshold

This property controls how many suggestions are returned from Atilika. Atilika gives each entity and concept it returns a relevance score between 0 (not relevant) and 1 (very

relevant). A **relevanceThreshold** setting of 0.3 rejects all suggestions that do not have a relevance score of 0.3 or higher.

connectTimeout

The interval after which a connection attempt is abandoned, specified in milliseconds.

requestTimeout

The interval after which a request for tag suggestions is abandoned, specified in milliseconds.

If you do not intend to make use of either entities or concepts, then you can omit the corresponding group of settings. If, for example you only intend to use entities, then you would only need to enter:

```
providers:
  - type: atilika_entities
    url: atilika-url
    relevanceThreshold: 0.3
    connectTimeout: 5000
    requestTimeout: 5000
    credentials:
      username: atilika-username
      password: atilika-password
```

4. Edit the following entries as required:

```
fields-to-analyze:
  - title
  - body

story-elements-to-analyze:
  - lead_text
  - paragraph
  - pull_quote
  - headline
```

fields-to-analyze determines which fields in classic rich text-based content items are sent to the external service for analysis. **story-elements-to-analyze** determines which story elements from storyline-based content items are sent to the external service for analysis.

5. By default, **cue-semantic-enrichment-service** runs on ports 9180 and 9181. If you need to, you can change this by modifying the following properties:

```
server:
  applicationConnectors:
    - type: http
      port: 9180

  adminConnectors:
    - type: http
      port: 9181
```

6. Save your changes.

2.2.1.2 Configuring the Feedback Service

You only need to configure this service if:

- You intend to make use of Atilika concepts
- You have installed the feedback service

The feedback service consists of two components:

- A **daemon** that monitors a publication for new events. This daemon is implemented using another Escenic product called the Change Log Daemon. For full information about this product and its capabilities, see [Change Log Daemon](#).
- An **agent** that selects concept tag creation events from the data collected by the daemon and returns information about them to the external service. Note that only information about concept tags is returned to the external service.

To configure **cue-semantic-feedback-service**:

1. Log in as **root**.
2. Open **/etc/escenic/cue-semantic-feedback-service/Daemon.properties** in an editor.
3. Add the following entries to the file:

```
urls = changelog-url
username = cue-username
password = cue-password
```

where:

- *changelog-url* is the URL of a publication change log. A publication change log URL looks like this:

```
http://content-store-host/webservice/escenic/changelog/
publication/publicationId
```

- *username* is a user name to be used for logging into the Content Engine and accessing the change log.
- *password* is the password for the specified user name.

The Change Log Daemon is actually capable of monitoring the change logs of more than one publication. If you need to be able to do that, see [Complex Configurations](#) to find out how.

4. Save your changes.
5. Create a file called **/etc/escenic/cue-semantic-feedback-service/Agent.properties** and open it in an editor.
6. Add the following entries to the file:

```
$class=com.escenic.semantic.atilika.FeedbackAgent
fieldsToAnalyze = title,body
storyElementsToAnalyze = lead_text,paragraph,pull_quote, headline
url= atilika-url
username = atilika-username
password = atilika-password
```



```
conceptsTagScheme = tag:topic@escenic.com,2017
```

Note that:

- The lists in **fieldsToAnalyze** and **storyElementsToAnalyze** must match the corresponding fields in `/etc/escenic/cue-semantic-enrichment-service/cue-semantic-enrichment-service.yaml` (see [section 2.2.1.1](#)).
- `atilika-url`, `atilika-username` and `atilika-password` must be access credentials supplied by Atilika.
- The tag scheme specified for **conceptsTagScheme** must be the one to which Atilika CON suggestions are mapped in `/etc/escenic/cue-web/config.yaml` (see [section 2.2.2](#)).

7. Save your changes.

2.2.2 Configuring CUE

To make CUE use CUE Semantic:

1. Log in as **root**.
2. Open `/etc/escenic/cue-web/public/Semantic.yaml` in an editor. This is what it contains (with all lines commented out).

```
enrichmentServices:
  - href: "http://semantic-service-host/"
    name: "Extract Semantic Entities"
    title: "Extract Semantic Entities"
    triggers:
      - name: editor-recurring
        properties:
          interval: 10
          timeout: 20
editors:
  metadata:
    - name: sp-escenic-semantic-panel
      directive: sp-escenic-semantic-panel
      cssClass: semanticplugin
      title: Semantic Analysis
      mimeTypees:
        - x-ece/story
        - x-ece/new-content; type=story
      args: ng-model="ngModel.resource.data" lock-helper="ngModel.lockHelper"
      requires:
        - escenic
      order: '850'
semanticEntityTypes:
  - name: PER
    tagScheme: tag:person@escenic.com,2017
    title: Persons
  - name: LOC
    tagScheme: tag:location@escenic.com,2017
    title: Locations
  - name: ORG
    tagScheme: tag:organization@escenic.com,2017
    title: Organizations
  - name: TTL
    tagScheme: tag:title@escenic.com,2017
    title: Titles
```

```
- name: FAC
  tagScheme: tag:facility@escenic.com,2017
  title: Facilities
- name: REL
  tagScheme: tag:religion@escenic.com,2017
  title: Religion
- name: NAT
  tagScheme: tag:nationality@escenic.com,2017
  title: Nationalities
- name: CON
  tagScheme: tag:concept@escenic.com,2017
  title: Concepts
```

3. Uncomment all the lines (or paste in the above content) and replace *semantic-service-host* with the host name or IP address of the host on which you installed CUE Semantic.
4. You may then want to modify the entries highlighted in **bold**:

interval

This property determines how frequently CUE Semantic requests new tag suggestions from the external service, specified in seconds.

timeout

This property determines how long CUE Semantic waits for a response from the external service before a "can't connect" message is displayed in the CUE Notification Center.

semanticEntityTypes

This entry contains an array of **name** / **tagScheme** / **title** properties that map the suggestions returned by the external service to Content Engine tag schemes. Atilika **concepts** are returned under the name **CON**. Atilika **entities** are divided into subcategories and returned under the following names:

- **PER** (persons such as Nelson Mandela, Donald Trump, etc.)
- **LOC** (locations such as Africa, Spain, New York, Chelsea)
- **ORG** (organizations such as companies, charities, political parties and so on)
- **TTL** (titles such as Mr, Dr, Sir etc.)
- **FAC** (facilities such as hotels, sports grounds, theatres, universities, shopping malls)
- **REL** (religions)
- **NAT** (nationalities such as Irish, Indian, Australian)

You can map each of these categories onto a separate tag scheme, or map multiple categories onto the same tag scheme if you wish, as is the case for **TTL**, **FAC**, **REL** and **NAT** in the example shown above. You can also choose to ignore a category by omitting a mapping for it. If, for example, you remove:

```
- name: TTL
  tagScheme: tag:entity@escenic.com,2017
```

```
title: Titles
```

from the example given above, then the **TTL** category will be ignored and no **title** tag suggestions will appear in CUE.

Concepts (**CON**), if used, should always be mapped to its own tag scheme that is not used for any other purpose. This is because tags in this tag scheme are the ones returned as feedback to the external service.

The tag schemes you reference in this section must already exist in the Content Engine. For information on how to create tag schemes, see [Manage Tag Structures](#).

5. Save your changes.

You will also probably want to change where the **Semantic Analysis** section appears in the metadata panel. You can control what sections appear in the metadata panel and the order in which they appear on a per-content type basis. For information on how to do this, see [Metadata Panel Sections](#). The name to use for the **Semantic Analysis** section is **sp-escenic-semantic-panel**, as defined in `/etc/escenic/cue-web/public/Semantic.yml` (above).

2.3 Running CUE Semantic

Once it has been installed and configured, CUE Semantic can be started as a service with the following **systemd** commands:

```
systemctl start cue-semantic-enrichment-service
systemctl enable cue-semantic-enrichment-service
systemctl start cue-semantic-feedback
systemctl enable cue-semantic-feedback
```

2.4 Troubleshooting

Both `cue-semantic-enrichment-service` and `cue-semantic-feedback-service` service maintain error logs:

```
/var/log/escenic/cue-semantic-enrichment-service.log
/var/log/escenic/cue-semantic-feedback-service.log
```

If one of the services is not running properly, check the appropriate log file for hints.

For the enrichment service, you can control logging levels by changing the settings in `/etc/escenic/cue-semantic-enrichment-service/cue-semantic-enrichment-service.yaml`:

```
logging:
  level: INFO
  loggers:
    io.dropwizard: INFO
    com.escenic.semantic : INFO
  appenders:
    - type: file
      currentLogFilename: /var/log/escenic/cue-semantic-enrichment-service.log
      archivedLogFilenamePattern: /var/log/escenic/cue-semantic-enrichment-service-
%d.log.gz
```

CUE Semantic User Guide

```
archivedFileCount: 5  
timeZone: UTC
```

For the feedback service, you can configure extra logging by adding a `$log` property to `/etc/escenic/cue-semantic-feedback-service/Agent.properties`:

```
$log = INFO
```

The feedback service also maintains a state file, `/var/lib/escenic/cue-semantic-feedback-service.state.properties`. This state file stores the current position in the change log. If you want to restart from the beginning, remove this file.

In the case where a service fails completely and doesn't even start, no log file will be generated. In this case, check the system log `/var/log/syslog`. This can happen, for example, if there is a typo in the enrichment service configuration file, `cue-semantic-enrichment-service.yaml`.

3 Using CUE Semantic

If CUE Semantic is installed and running, then a **Semantic Analysis** section is added to the CUE content editor's metadata panel, and used to display a list of suggested tags for the content currently displayed in the editor:

The list of suggestions is constantly updated to reflect changes in the content. To tag a content item, all you have to do is select suggestions from this list by clicking on them. When you select a suggestion, it is grayed out to indicate that it is now selected, and it immediately appears in the **Tags** section of the metadata panel as a tag. Once it is added here as a tag, you can optionally adjust the relevance score suggested by the external service.

When a suggestion is accepted, one of two things can happen:

- If the suggestion **exactly** matches the label of an existing tag in the tag schema to which the suggestion has been mapped, then that tag is used to tag the content item.
- If the suggestion does not exactly match the label of an existing tag in the tag schema to which the suggestion has been mapped, then a new tag is added to the tag schema.

Note that:

- The tag label matching is case sensitive. If your tag schema contains a tag called **football**, accepting a suggestion with the name **Football** will result in the creation of a new tag.
- New tags are always added at the root level of a tag schema.

For these reasons, it may be necessary to periodically tidy up your tag schemas by merging tags and moving them from the root level to the appropriate location in tree-structured tag schemas, especially if you use CUE Semantic to update existing tag schemas that contain tags not suggested by CUE Semantic's external service.

CUE Semantic tries to encourage consistent tagging by:

- Marking suggested tags that do not already exist in the tag schema as new (they are marked with a triangular tab as shown in the screenshot above)
- Asking for confirmation when you add a new tag.

Before adding a new tag, you should always check to see if there is another semantically equivalent tag that already exists in the tag schema. In this way you will avoid the need for merging tags later.